Example

This is the notebook to illustrate the usage of the `ZCAPM` package.

List of existing functions and dataset in the package

```
library(ZCAPM)
ls("package:ZCAPM")
```

```
##  [1] "CheckPackage"   "EM_loop"        "EMRegression"   "estLM"
##  [5] "estZCAPM"       "FactorModel"    "ff_factors"     "ff25_day"
##  [9] "fm_test"        "GetMonthReturn" "ind47"          "MonthRoll"
## [13] "mu_sigma"       "print.summary"  "RollForward"    "SingleReg"
## [17] "ZCAPM"
```

- `ff25_day`, `ind47`, `ff_factors`, and `mu_sigma` are the four datasets we included in this package as examples. The first two are datasets for 25 size-B/M (book-to-market equity) sorted portfolios and 47 industry portfolios from January 1964 to December 2015, while the rest two store the factors can be considered in the model.
- `GetMonthReturn` can be used to get monthly return according to daily return.
- `EMRegression`, `estLM` and `FactorModel` are three functions used for traditional models, e.g. CAPM market model, Fama and French's three-factor model, Carhart's four-factor model, and Fama and French's five-factor model. You can design your own factor model as well.
- `EM_loop`, `estZCAPM` and `ZCAPM` are three functions used for our proposed ZCAPM with EM algorithm.
- `SingleReg` and `fm_test` are the functions used for the two-step Fama and MacBeth (1973) procedure for cross-sectional tests.
- `MonthRoll` and `RollForward` are the functions for month-rolling estimation and apply a certain function to a data in a rolling window style.

For more detailed information, please use `help()` for each function of interest.

Example for illustration

Load the dataset and data preprocessing

```
data("SizedBM25") # 25 size B/M portfolios
data("mu_sigma")
data("ff_factors")

# select portfolio excess return for period of interest
# get factor data for period of interest
start.date <- 19640101
end.date <- 20151231
data.port <- subset(ff25_day, Date >= start.date & Date <= end.date)
```

```r
data.factor <- subset(mu_sigma, Date >= start.date & Date <= end.date)
data.factor1 <- subset(ff_factors, Date >= start.date & Date <= end.date)
data.factor$SMB <- data.factor1$SMB
data.factor$HML <- data.factor1$HML
rm(data.factor1)

# check if the dates of portfolios and covariates matched
stopifnot(all(data.port$Date == data.factor$Date))

# get year and month
data.yyyymm <- as.numeric(substr(data.port$Date, 1, 6))
month.list <- sort(unique(data.yyyymm))

# get excess return of each portfolio
data.port <- data.port[, -1]
data.port.exc.ret <- as.data.frame(apply(data.port, 2,
                                         FUN = function(x, y){x - y}, data.factor$R_f))

# get monthly excess return of each portfolio
port.exc.ret.month <- aggregate(data.port.exc.ret,
                                by = list(yyyymm = data.yyyymm), FUN = GetMonthReturn)[, -1]
```

Set the parameters

```r
# set rolling window, unit: month
roll.width <- 12
tol <- 0.001
MaxIter <- 1000

# criterion = 1: if 'abs(diff(new - old)/old)' is used as convergence criterion
# criterion = 2: if 'abs(diff(new - old)/(abs(old) + 1))' is used as the convergence criterion
# criterion = 2 usually leads to better convergence
criterion <- 1
params <- c(tol, MaxIter, criterion)

# keep only the one-month ahead monthly return in testing period
port.exc.ret.month.test <- port.exc.ret.month[-(1:roll.width), ]

# Specify the months used for model estimation
month.list.est <- month.list[-length(month.list)]
```

Estimation using factor model and ZCAPM

```r
# zcapm model
zcapm.res <- ZCAPM(data.port.exc.ret, month.list.est, roll.width,
                   data.yyyymm, data.factor[, c("R_a.R_f", "sigma_a")], params)
```

```
## The time period:  198308 - 198407
## Excess the MaxIter: 1001
## The time period:  198909 - 199008
## Excess the MaxIter: 1001
```

```
## The time period:  198910 - 199009
## Excess the MaxIter: 1001
## The time period:  197203 - 197302
## Excess the MaxIter: 1001
## The time period:  197109 - 197208
## Excess the MaxIter: 1001
## The time period:  197110 - 197209
## Excess the MaxIter: 1001
## The time period:  197707 - 197806
## Excess the MaxIter: 1001
```

```
# capm and 3-factor model
capm.res <- FactorModel(data.port.exc.ret, month.list.est, roll.width,
                        data.yyyymm, data.frame(R_a.R_f = data.factor[, "R_a.R_f"]))
ff3.res <- FactorModel(data.port.exc.ret, month.list.est, roll.width,
                       data.yyyymm, data.factor[, c("R_a.R_f", "SMB", "HML")])

# adjustment to the zeta's of zcapm
num.mod <- nrow(zcapm.res$factor.loading) / 2
zcapm.res$factor.loading[(num.mod + 1): (num.mod * 2), ] <- 21 *
                         zcapm.res$factor.loading[(num.mod + 1): (num.mod * 2), ]
```

Out-of-sample cross-sectional Fama and MacBeth test

```
# Specify the list of models to be tested
# The name of variables stand for the models
# this will be the name of model show in the summary table
fact.load.lst <- list(ZCAPM = zcapm.res,
                      LM = capm.res,
                      FF3 = ff3.res)

# run the FM-test
test_res <- fm_test(port.exc.ret.month.test, fact.load.lst)

# print out the summary table
print.summary(test_res)
```

```
##
## Model:  ZCAPM
## Variables Included:  R_a.R_f sigma_a
## Number of periods includes: 612
##
## Monthly rolling approach:
##            mean.coef  t.value
## Intercept  0.7593263  3.08757
## R_a.R_f   -0.1783318 -0.72660
## sigma_a    0.4885823  4.29995
##
## Single regression approach:
## R-squared:  0.969061
##
## = = = = = = = = = = = = = =
##
```

```
## Model:  LM
## Variables Included:  R_a.R_f
## Number of periods includes: 612
##
## Monthly rolling approach:
##            mean.coef  t.value
## Intercept  0.9235086  3.75503
## R_a.R_f   -0.2995732 -1.19801
##
## Single regression approach:
## R-squared:  0.5240715
##
## = = = = = = = = = = = = =
##
## Model:  FF3
## Variables Included:  R_a.R_f SMB HML
## Number of periods includes: 612
##
## Monthly rolling approach:
##           mean.coef  t.value
## Intercept  0.899905  4.68343
## R_a.R_f   -0.382649 -1.82572
## SMB        0.180637  1.36320
## HML        0.300132  2.53080
##
## Single regression approach:
## R-squared:  0.649313
```