Skill and Luck in Performance Evaluation -Fama/French (2010)

Taisei Noda GitHub Link

In [1]:
```
using CSV, Plots,JuMP, Random, Statistics, Bootstrap, Distributions, LinearAlgebra,Dates,DataFrames,Gurobi,DataFram

function estimateFF(lambda,alpha;N = 1000, T = 120,B = 100,market_return_mean = 0.05/12,market_return_sd = 0.2/sqrt
    market_return_t = randn(T).*market_return_sd.+market_return_mean
    market_return = zeros(N,T)
    for i = 1:N
        market_return[i,:] = market_return_t[:]
    end
    residual = randn(N,T).*residual_sd
    if lambda == 0.0
        true_alphas = zeros(N,T)
    elseif lambda > 0.0
        true_alphas = zeros(N,T)
        for i = 1:floor(Int64,N*lambda)
                true_alphas[i,:] .= alpha/12
        end
    end
    fund_return = true_alphas.+market_return.+residual #beta = 1
    estimator = zeros(N,2)
    est_residual = zeros(N,T)
    t_stats_cross = zeros(N)
    for n = 1:N
        y = fund_return[n,:]
        X = hcat(ones(T),market_return[n,:])
        estimator[n,:] = (X'X)\(X'y)
        est_residual[n,:] = y - X*estimator[n,:]
        ssr = sum((y - X*estimator[n,:]).^2)/length(y)
        var = ssr*inv(X'*X)
        t_stats_cross[n] = estimator[n,1]/sqrt(var[1,1])
    end
    bs_estimator = zeros(N,2)
    bs_alphas = zeros(N,B)
    bs_t_stats = zeros(N,B)
    for b = 1:B
        bootstrap_ts = sample((1:T),T,replace=true)
        bs_market_return = zeros(N,length(bootstrap_ts))
        bs_fund_return = zeros(N,length(bootstrap_ts))
        for t = 1:length(bootstrap_ts)
            bs_market_return[:,t] = market_return[:,bootstrap_ts[t]]
            bs_fund_return[:,t] = true_alphas[:,t] + market_return[:,bootstrap_ts[t]].*estimator[:,2]
        end
        for n = 1:N
            X = hcat(ones(T),bs_market_return[n,:])
            y = bs_fund_return[n,:]
            bs_estimator[n,:] = (X'X)\(X'y)
            ssr = sum((y - X*estimator[n,:]).^2)/length(y)
            var = ssr*inv(X'*X)
            bs_alphas[n,b] = bs_estimator[n,1]
            bs_t_stats[n,b] = bs_estimator[n,1]/sqrt(var[1,1])
        end
    end
    bs_t_sort = zeros(N,B)
    bs_t_average = zeros(N)
    bs_t_5 = zeros(N)
    bs_t_95 = zeros(N)
    for n = 1:N
        bs_t_sort[n,:] = sort(bs_t_stats[n,:])
        bs_t_average[n] = sum(bs_t_stats[n,:])/length(bs_t_stats[n,:])
        bs_t_5[n] = bs_t_sort[n,floor(Int64,B*0.05)]
        bs_t_95[n] = bs_t_sort[n,floor(Int64,B*0.95)]
    end
    h1 = histogram(t_stats_cross,xaxis = ((-10, 10),-10:2:10),yaxis=((0,300),0:50:300), xlabel="T-stat cross-sectio
    h2 = histogram(bs_t_average,xaxis = ((-10, 10),-10:2:10),yaxis=((0,300),0:50:300),xlabel="Bootstrap Average",le
    h3 = histogram(bs_t_5,xaxis = ((-10, 10),-10:2:10),yaxis=((0,300),0:50:300),xlabel="Bootstrap 5 percentile",leg
    h4 = histogram(bs_t_95,xaxis = ((-10, 10),-10:2:10),yaxis=((0,300),0:50:300),xlabel="Bootstrap 95 percentile",l
    l = @layout [a{.1h};grid(2,2)]
    hist = plot(plot(annotation=(0.5,0.5,text("Lambda = $lambda, Alpha = $alpha",:center,48)),framestyle = :none),h
    return hist
end
```

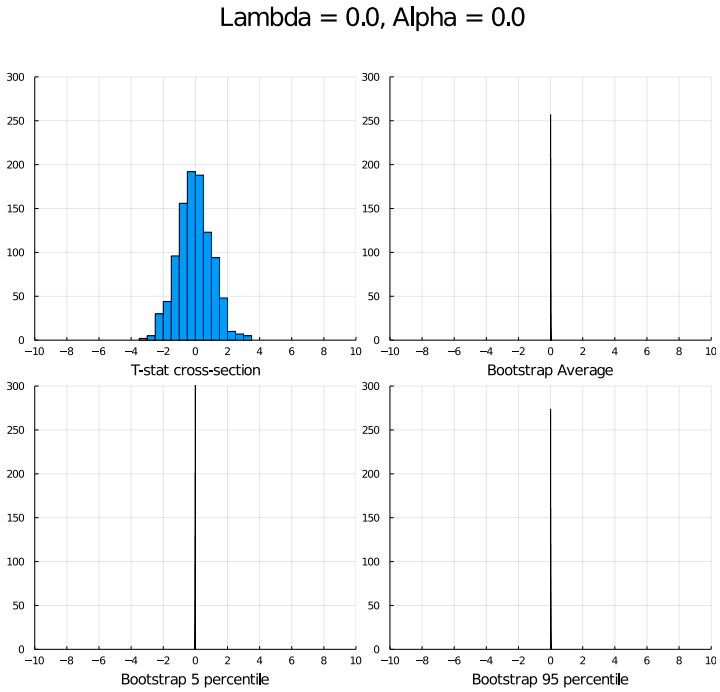
Out[1]:
```
estimateFF (generic function with 1 method)
```

Part 1- No Skilled Funds

In [2]:
```
Plots.scalefontsizes(2.5)
```

In [3]:
```
res_no_skill = estimateFF(0.0,0.0)
res_no_skill
```

Out[3]:

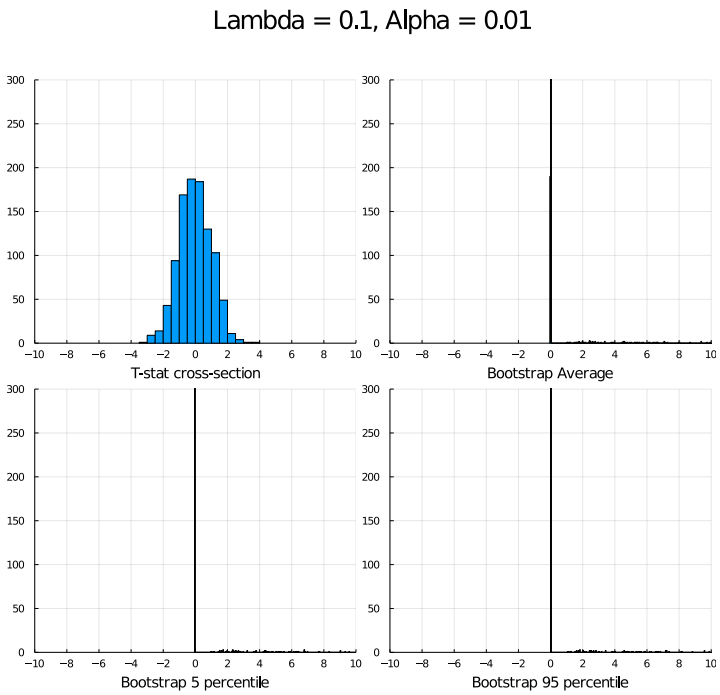


Part 2- Some Skilled Funds

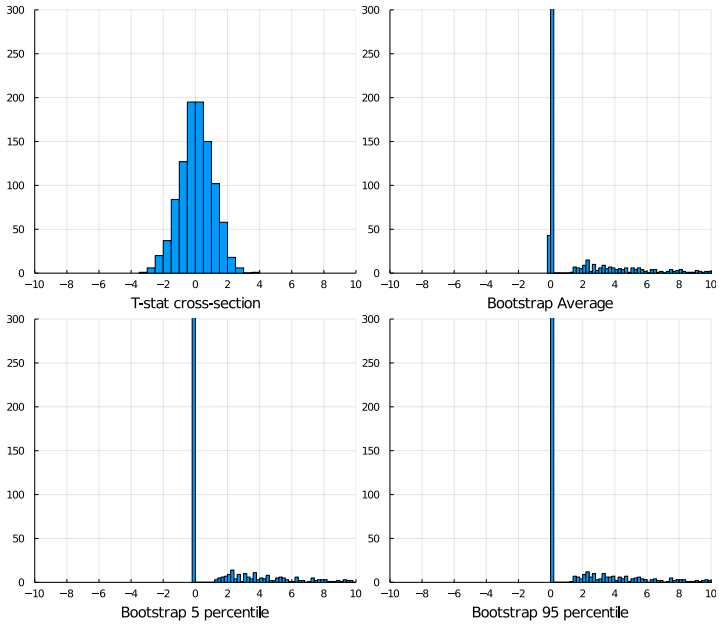
In [4]:
```
using IJulia
```

In [5]:
```
Lambdas = [0.1,0.25,0.5,0.75];
Alphas = [0.01,0.025,0.05];
```

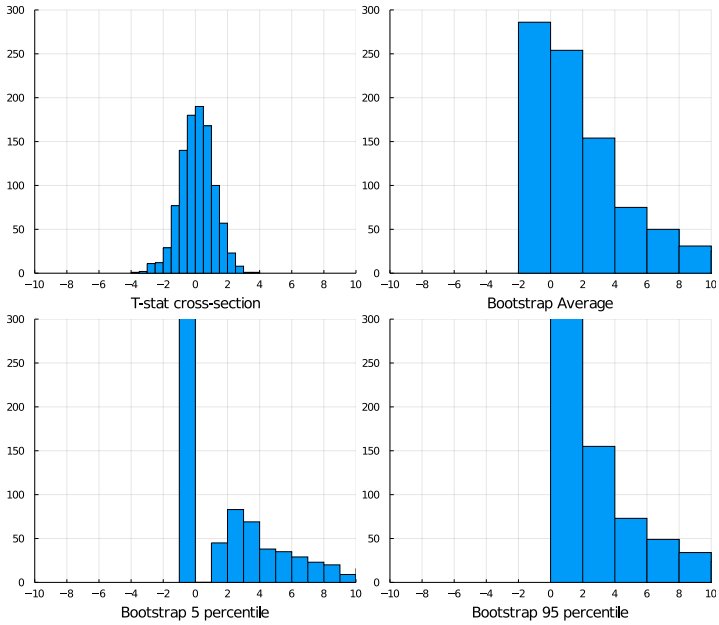
In [6]:
```
for a = 1:length(Alphas)
    for l = 1:length(Lambdas)
        Plots.display(estimateFF(Lambdas[l],Alphas[a]))
    end
end
```



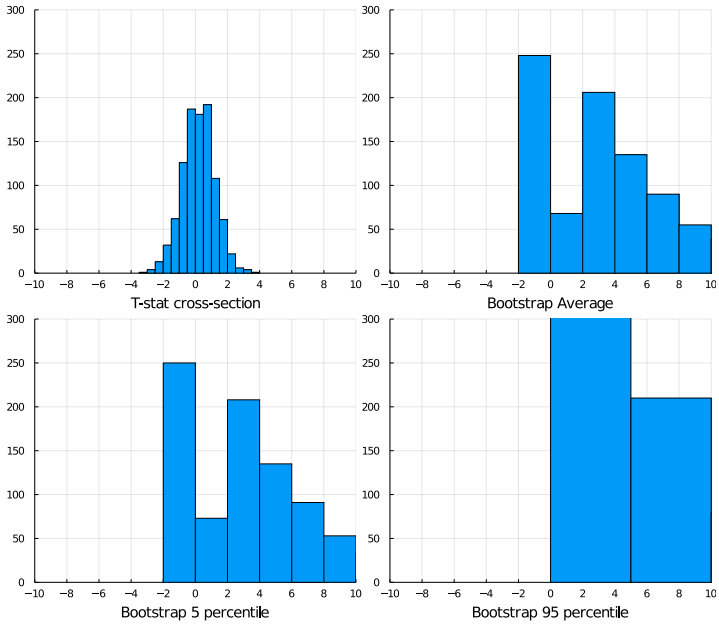
Lambda = 0.25, Alpha = 0.01



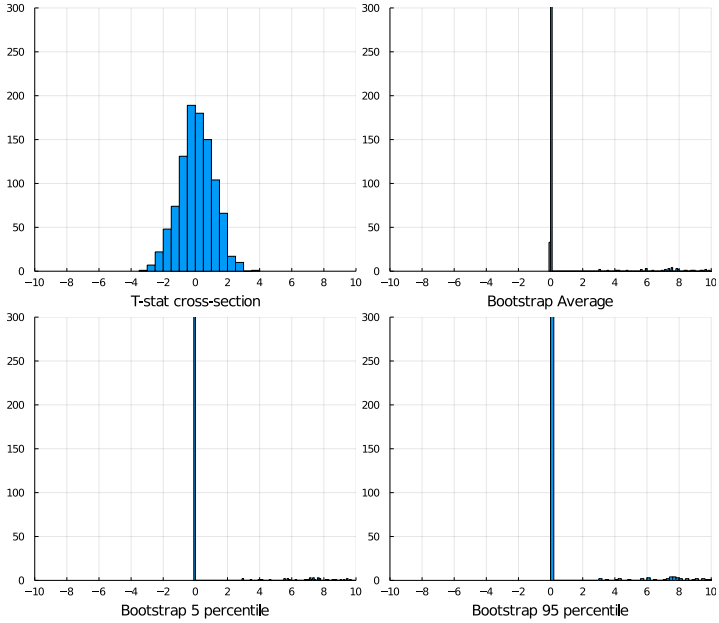
Lambda = 0.5, Alpha = 0.01



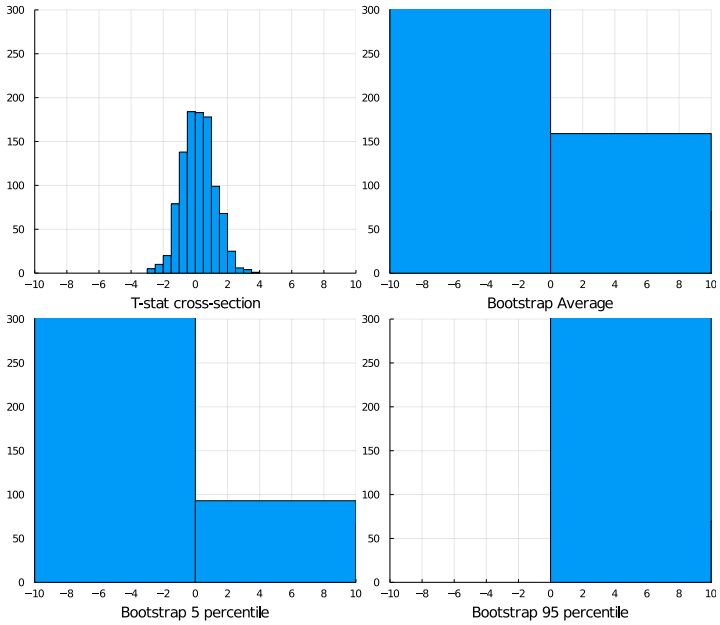
Lambda = 0.75, Alpha = 0.01



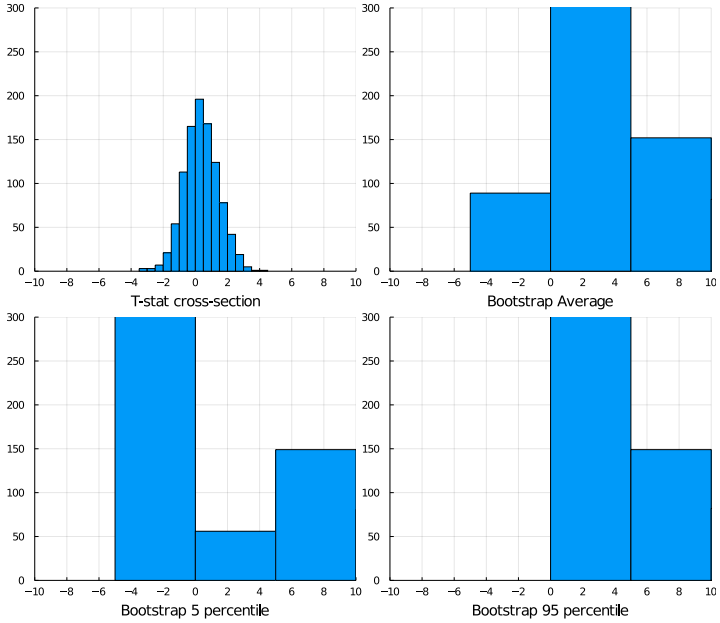
Lambda = 0.1, Alpha = 0.025



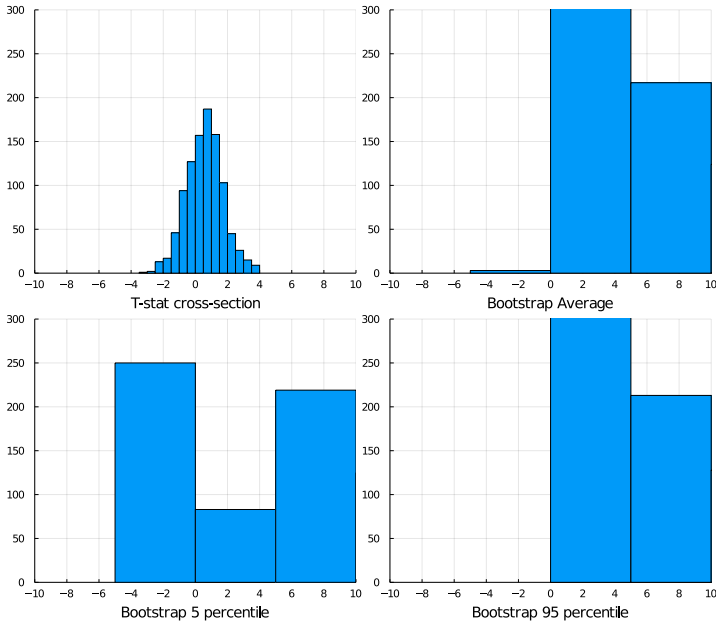
Lambda = 0.25, Alpha = 0.025



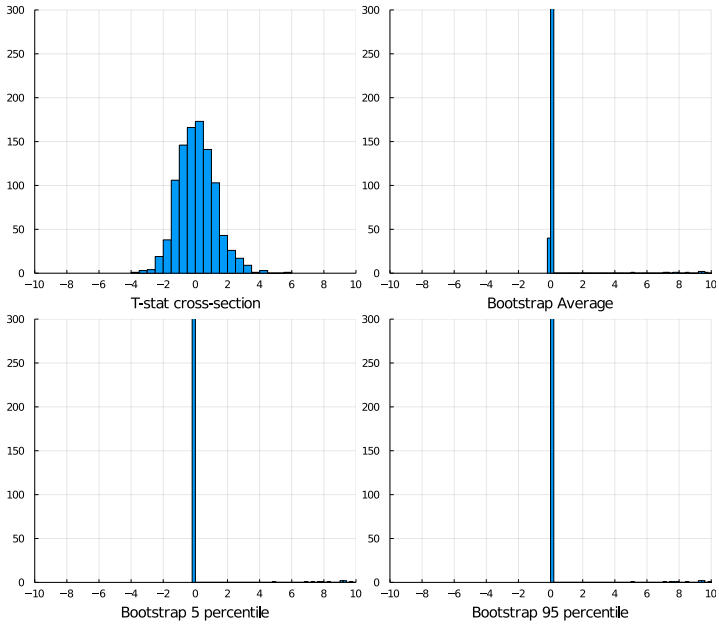
Lambda = 0.5, Alpha = 0.025



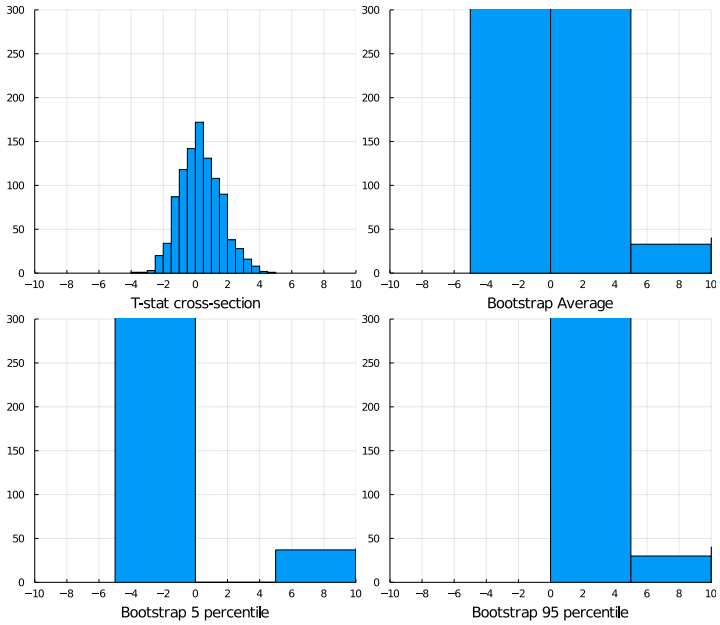
Lambda = 0.75, Alpha = 0.025



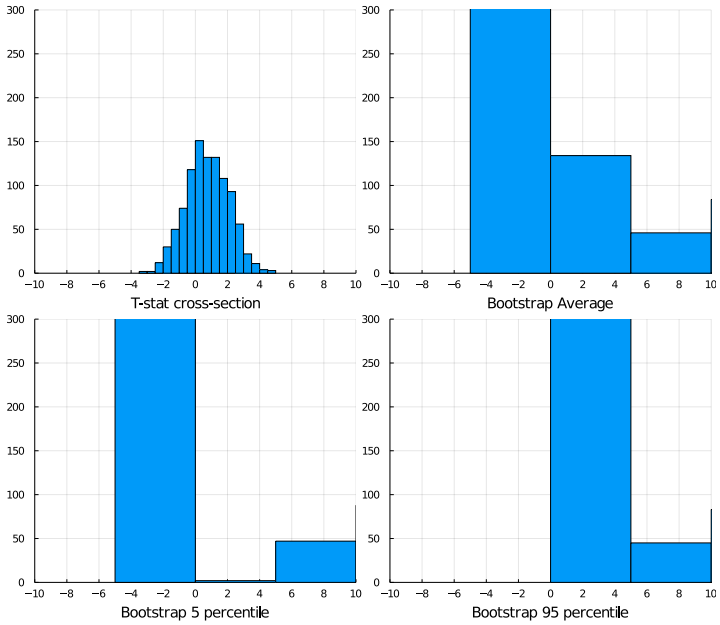
Lambda = 0.1, Alpha = 0.05

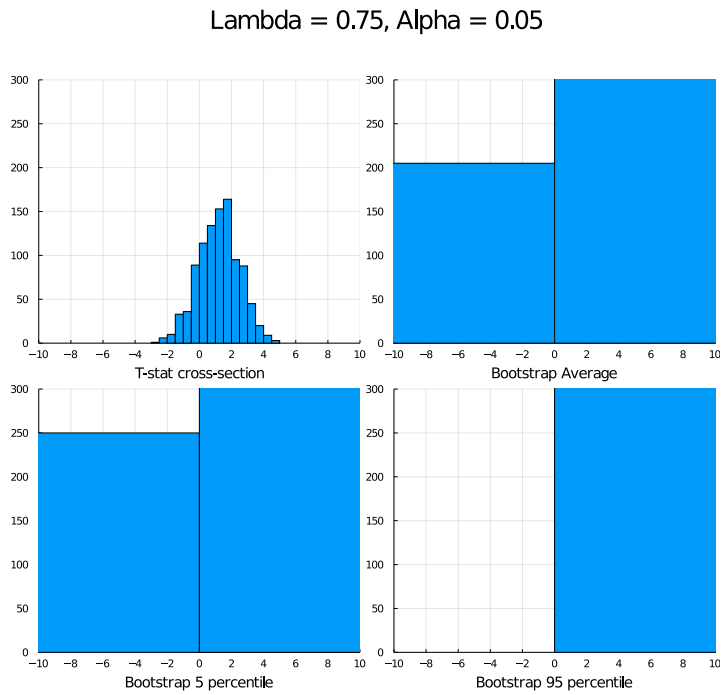


Lambda = 0.25, Alpha = 0.05



Lambda = 0.5, Alpha = 0.05




Lambda = 0.75, Alpha = 0.05
T-stat cross-section
Bootstrap Average
Bootstrap 5 percentile
Bootstrap 95 percentile

In [ ]: