

# BUSI 525 PS1 "Skill or Luck"

Taisei Noda

[Github link](#)

```
In [1]: using CSV, Plots, JuMP, Random, Distributions, LinearAlgebra, Dates, DataFrames, Gurobi, DataFramesMeta
using LinearRegression

function skill_tstat(lambda; N = 1000, T = 120, market_return_mean = 0.05/12, market_return_sd = 0.2/sqrt(12), resid

    market_return = randn(N,T).*market_return_sd.+market_return_mean
    residual = randn(N,T).*residual_sd
    if lambda == 0.0
        alpha = zeros(N,T)
    elseif lambda > 0.0
        alpha = zeros(N,T)
        for i = 1:floor(Int64,N*lambda)
            alpha[i,:] = 0.05
        end
    end
    alpha_vec = alpha[:,1]
    fund_return = alpha.*market_return.+residual
    estimator = zeros(N,2)
    constant = ones(T)
    t_stats = zeros(N)
    p_values = zeros(N)
    T_Dist = TDist(T)
    for n = 1:N
        x = market_return[n,:]
        y = fund_return[n,:]
        X = hcat(constant,x)
        estimator[n,:] = (X'X)\(X'y)
        ssr = sum((y - X*estimator[n,:]).^2)/length(y)
        var = ssr*inv(X'*X)
        t_stats[n] = estimator[n,1]/sqrt(var[1,1])
        p_values[n] = 2*(1-cdf(T_Dist,abs(t_stats[n])))
    end
    println("Lambda = ",lambda)
    println("The ratio of skilled funds = ",sum(t_stats.>1.96)/N)
    println("The ratio of true positive = ",sum((t_stats.>1.96).*(alpha_vec.>0.0))/N)
    println("The ratio of false negative = ",sum((t_stats.<1.96).*(alpha_vec.>0.0))/N)
    println("The ratio of true negative = ",sum((t_stats.<1.96).*(alpha_vec.==0.0))/N)
    println("The ratio of false positive = ",sum((t_stats.>1.96).*(alpha_vec.==0.0))/N)
    println(" ")
    println(" ")
    println(" ")

    return results = t_stats, p_values
end

L = [0.0,0.1,0.25,0.5,0.75]
tstat_results = zeros(length(L),1000)
pvalue_results = zeros(length(L),1000)
for l = 1:length(L)
    tstat_results[l,:],pvalue_results[l,:] = skill_tstat(L[l])
end
```

```
Lambda = 0.0
The ratio of skilled funds = 0.033
The ratio of true positive = 0.0
The ratio of false negative = 0.0
The ratio of true negative = 0.967
The ratio of false positive = 0.033
```

```
Lambda = 0.1
The ratio of skilled funds = 0.127
The ratio of true positive = 0.1
The ratio of false negative = 0.0
The ratio of true negative = 0.873
The ratio of false positive = 0.027
```

```
Lambda = 0.25
The ratio of skilled funds = 0.272
The ratio of true positive = 0.25
The ratio of false negative = 0.0
The ratio of true negative = 0.728
```

The ratio of false positive = 0.022

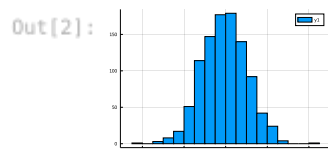
Lambda = 0.5  
The ratio of skilled funds = 0.508  
The ratio of true positive = 0.5  
The ratio of false negative = 0.0  
The ratio of true negative = 0.492  
The ratio of false positive = 0.008

Lambda = 0.75  
The ratio of skilled funds = 0.761  
The ratio of true positive = 0.75  
The ratio of false negative = 0.0  
The ratio of true negative = 0.239  
The ratio of false positive = 0.011

## Distribution of t-stats

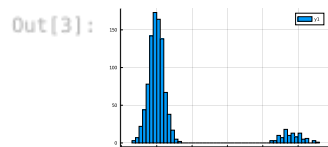
- $\lambda = 0.0$

```
In [2]: histogram(tstat_results[1,:])
```



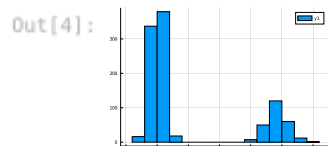
- $\lambda = 0.1$

```
In [3]: histogram(tstat_results[2,:])
```



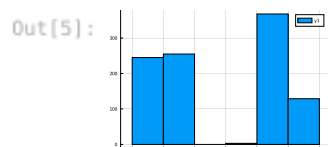
- $\lambda = 0.25$

```
In [4]: histogram(tstat_results[3,:])
```



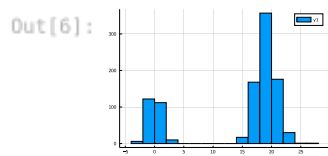
- $\lambda = 0.5$

```
In [5]: histogram(tstat_results[4,:])
```



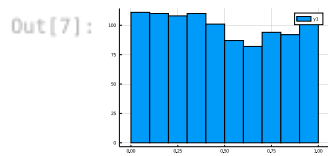
- $\lambda = 0.75$

In [6]: `histogram(tstat_results[5,:])`



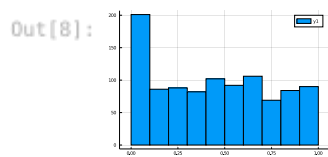
- $\lambda = 0.0$

In [7]: `histogram(pvalue_results[1,:])`



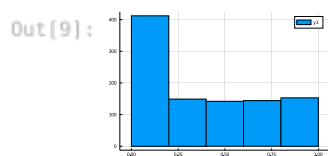
- $\lambda = 0.1$

In [8]: `histogram(pvalue_results[2,:])`



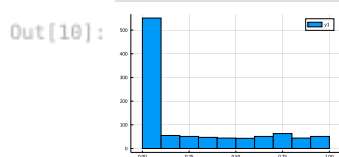
- $\lambda = 0.25$

In [9]: `histogram(pvalue_results[3,:])`



- $\lambda = 0.5$

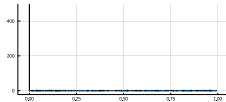
In [10]: `histogram(pvalue_results[4,:])`



- $\lambda = 0.75$

In [12]: `histogram(pvalue_results[5,:])`





In [ ]:

Processing math: 100%