Correlations:

The Tango of Time Series

Lance Hester
Learning Tuesday
06/02/2020

https://www.netclipart.com/isee/iRmohTm_ballroom-dancing-tango-art/

What you Should Take Away Today

Getting to know your time series by:

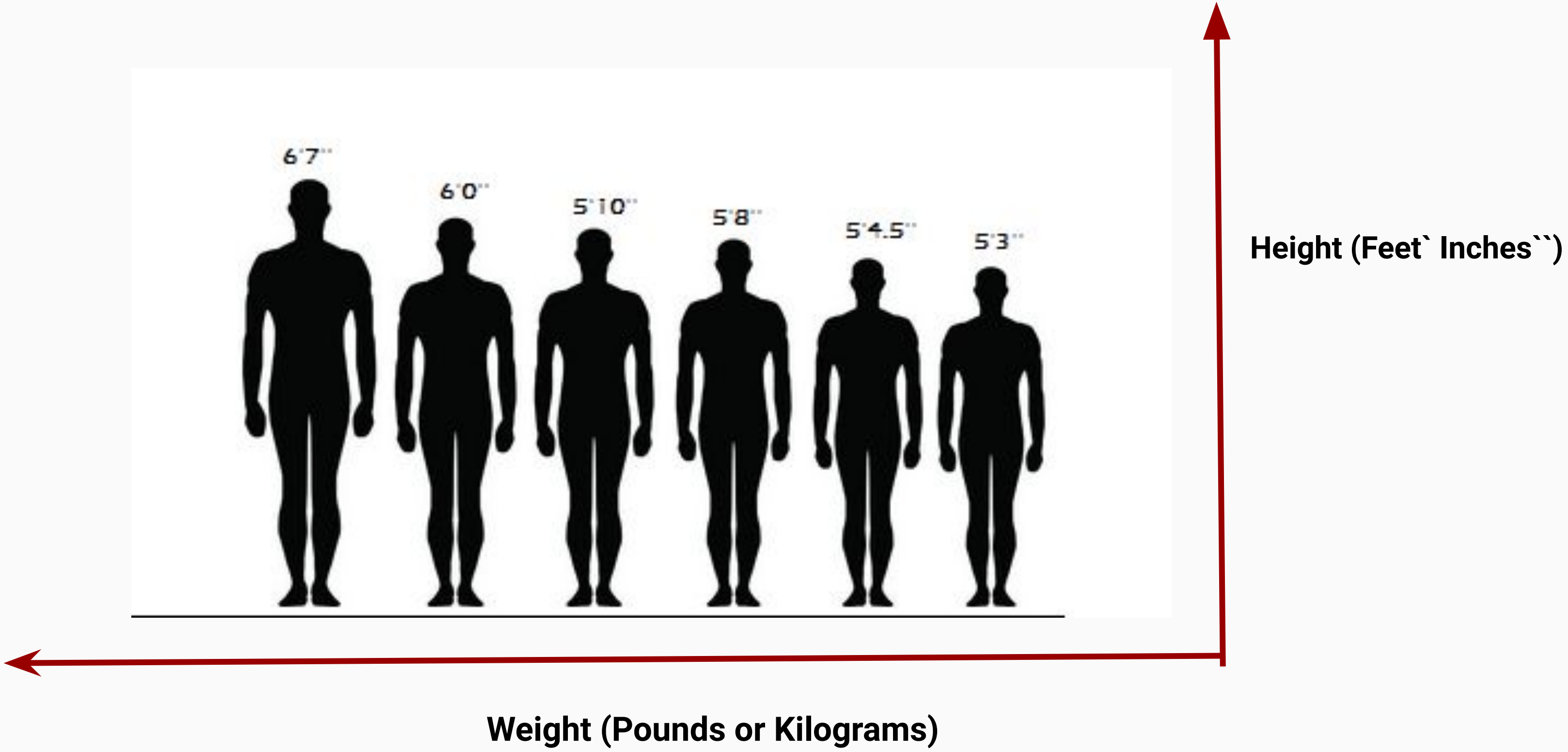
- Comparing it to itself (auto = “self”)
- Comparing other time series (cross)

Correlation

Definition *cor·re·la ·tion*

The degree of correspondence or relationship between two variables.

Correlation of Two Metrics/Time Series



Formula Flavors

Pearson (parametric)

r

Kendall Rank (non-parametric)

τ

Spearman Correlation (non-parametric)

ρ

Pearson Correlation Formula

$$\boxed{\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathrm{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}}$$

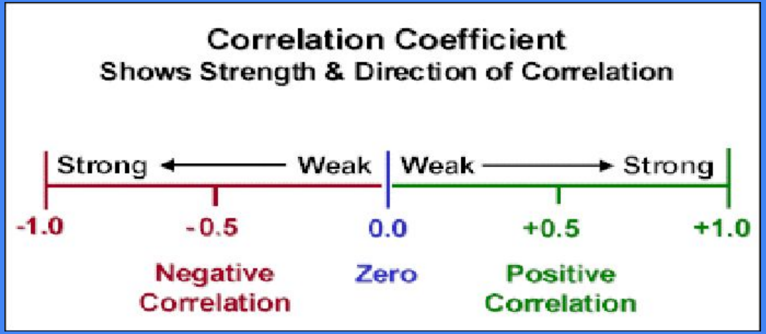
$$\boxed{\rho_{X,Y} = \frac{\mathrm{E}(XY) - \mathrm{E}(X)\,\mathrm{E}(Y)}{\sqrt{\mathrm{E}(X^2) - \mathrm{E}(X)^2} \cdot \sqrt{\mathrm{E}(Y^2) - \mathrm{E}(Y)^2}}}$$

https://en.wikipedia.org/wiki/Kendall_rank_correlation_coefficient

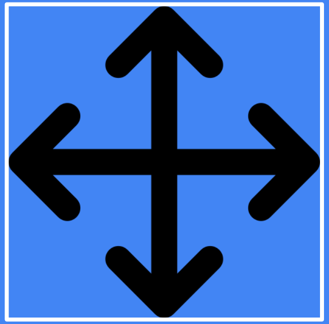
Correlation Coefficient
r

(quantitative measure)
(aka, a number!)

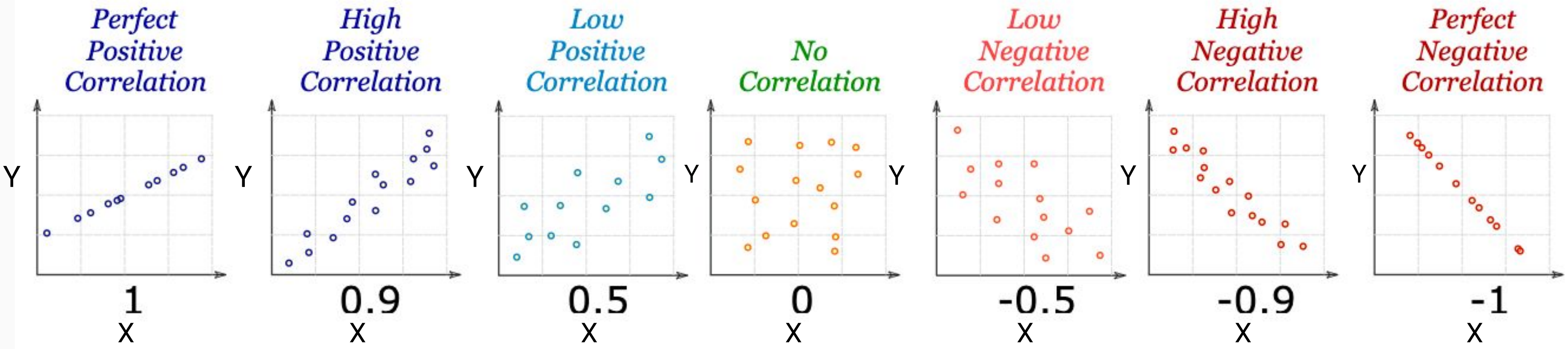
https://www.researchgate.net/figure/Strength-of-Coefficient-of-Correlation_fig1_272863721



Strength



Direction



https://www.mathsisfun.com/definitions/correlation.html

Thanks!

https://anomaly.io/index.html

Provided good examples of Correlations which I Julia-ized and added some additional information.

https://anomaly.io/detect-anomalies-in-correlated-time-series/index.html

https://anomaly.io/understand-auto-cross-correlation-normalized-shift/index.html#/cross_correlation

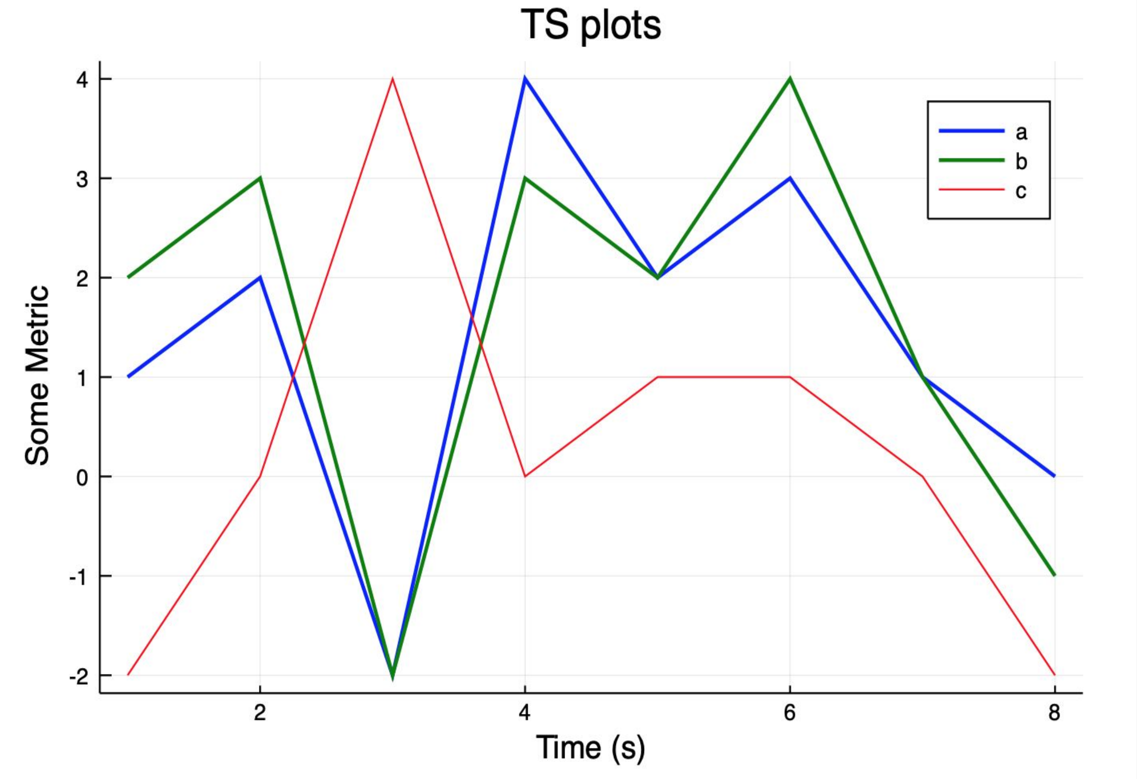
https://anomaly.io/detect-correlation-time-series/index.html

Cross-Correlation:

It Takes Two!

```
a = [1,2,-2,4,2,3,1,0];
b = [2,3,-2,3,2,4,1,-1];
c = [-2,0,4,0,1,1,0,-2];
```

$$corr(x, y) = \sum_{n=0}^{n-1} x[n] * y[n]$$



$$\begin{aligned} corr(a, b) &= 1 * 2 + 2 * 3 + -2 * -2 + 4 * 3 + 2 * 2 + 3 * 4 + 1 * 1 + 0 * -1 \\ &= 41 \end{aligned}$$

$$\begin{aligned} corr(a, c) &= 1 * -2 + 2 * 0 + -2 * 4 + 4 * 0 + 2 * 1 + 3 * 1 + 1 * 0 + 0 * -2 \\ &= -5 \end{aligned}$$

Issues with Raw Cross-Correlation Calculations

1. Can't really grasp value of cross_ab vs cross_ac significance
2. Want a&b or a&c to have similar amplitudes - might misread correlations.

$$\begin{aligned} corr(a, a/2) &= 1 * (1/2) + 2 * (2/2) + -2 * (-2/2) + 4 * (4/2) + 2 * (2/2), 3 * (3/2) + 1 * (1/2) + 0 * (0/2) \\ &= 19.5 \end{aligned}$$

3. Have to ensure that std deviation values are finite and positive.

“Solution: Normalize the Values”

Normalization alleviates these issues so we can compare.

$$norm\_corr(x,y) = \frac{\sum_{n=0}^{n-1} x[n] * y[n]}{\sqrt{\sum_{n=0}^{n-1} x[n]^2 * \sum_{n=0}^{n-1} y[n]^2}}$$

$$norm\_corr(x,y) = PearsonCorrelationCoefficient$$

Using this formula let's compute the normalized cross-correlation of ab and ac.

$$\begin{aligned} norm\_corr(a,b) &= \frac{1*2+2*3+-2*-2+4*3+2*2+3*4+1*1+0*-1}{\sqrt{(1+4+4+16+4+9+1+0)*(4+9+4+9+4+16+1+1)}} \\ &= \frac{41}{\sqrt{(39)*(48)}} \\ &= 0.947 \end{aligned}$$

$$\begin{aligned} norm\_corr(a,c) &= \frac{1*-2+2*0+-2*4+4*0+2*1+3*1+1*0+0*-2}{\sqrt{(1+4+4+16+4+9+1+0)*(4+0+16+0+1+1+0+4)}} \\ &= \frac{-5}{\sqrt{(39)*(26)}} \\ &= -0.157 \end{aligned}$$

```
norm_corr_ab = sum(a .* (b)) / sqrt(sum(a.^2) * sum(b.^2)); # equals  0.947
norm_corr_ac = sum(a .* (c)) / sqrt(sum(a.^2) * sum(c.^2)); # equals -0.157
```

Quick Check to Show Normalization Works

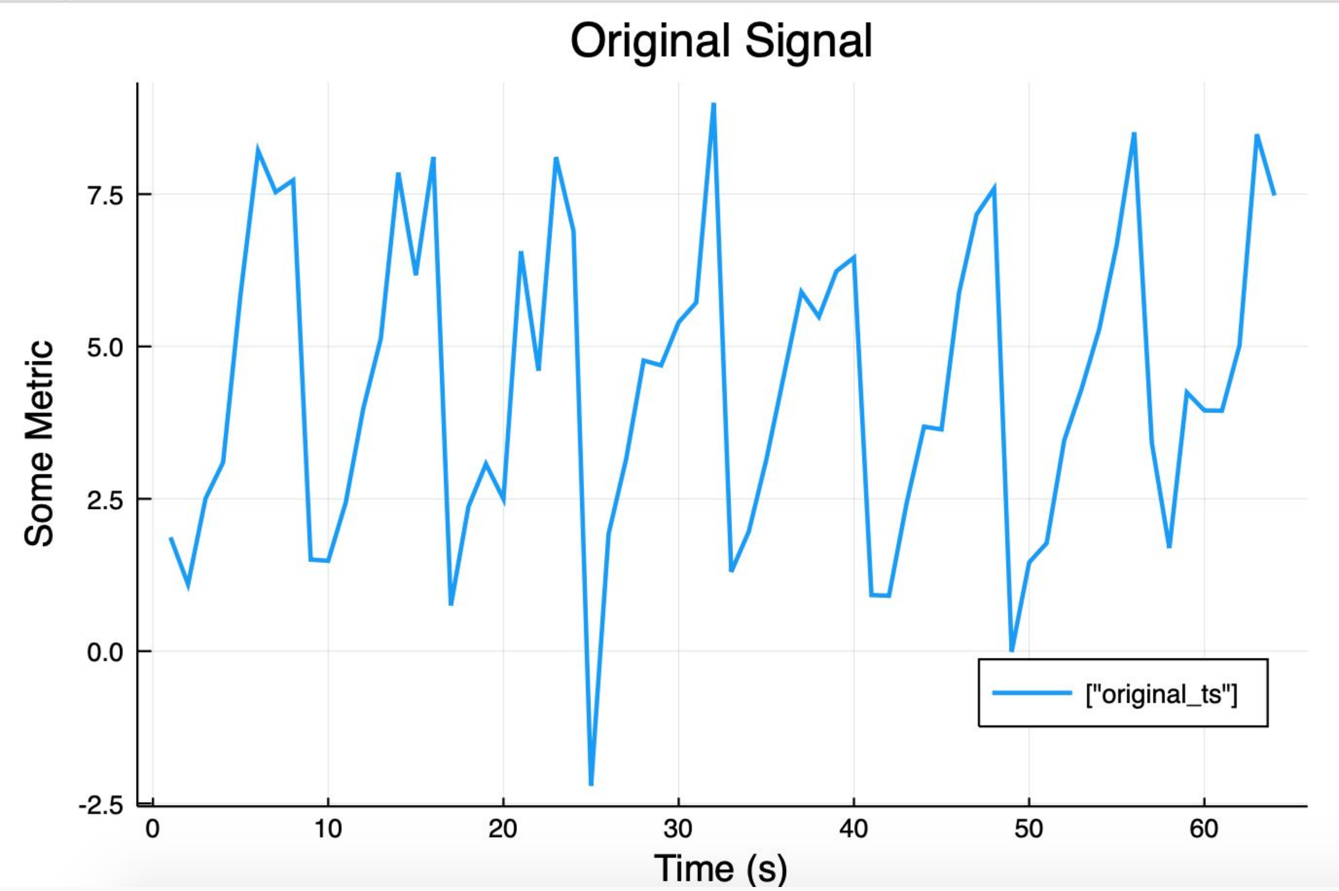
In [4]:
```
# Normalized norm_corr(a,a) = 1:
proof = sum(a .* (a)) / sqrt(sum(a.^2) * sum(a.^2))
```
Out[4]: 1.0

In [5]:
```
# Normalized norm_corr(a,-a) = -1:
proof = sum(a .* (-a)) / sqrt(sum(a.^2) * sum(((-a).^2)))
```
Out[5]: -1.0

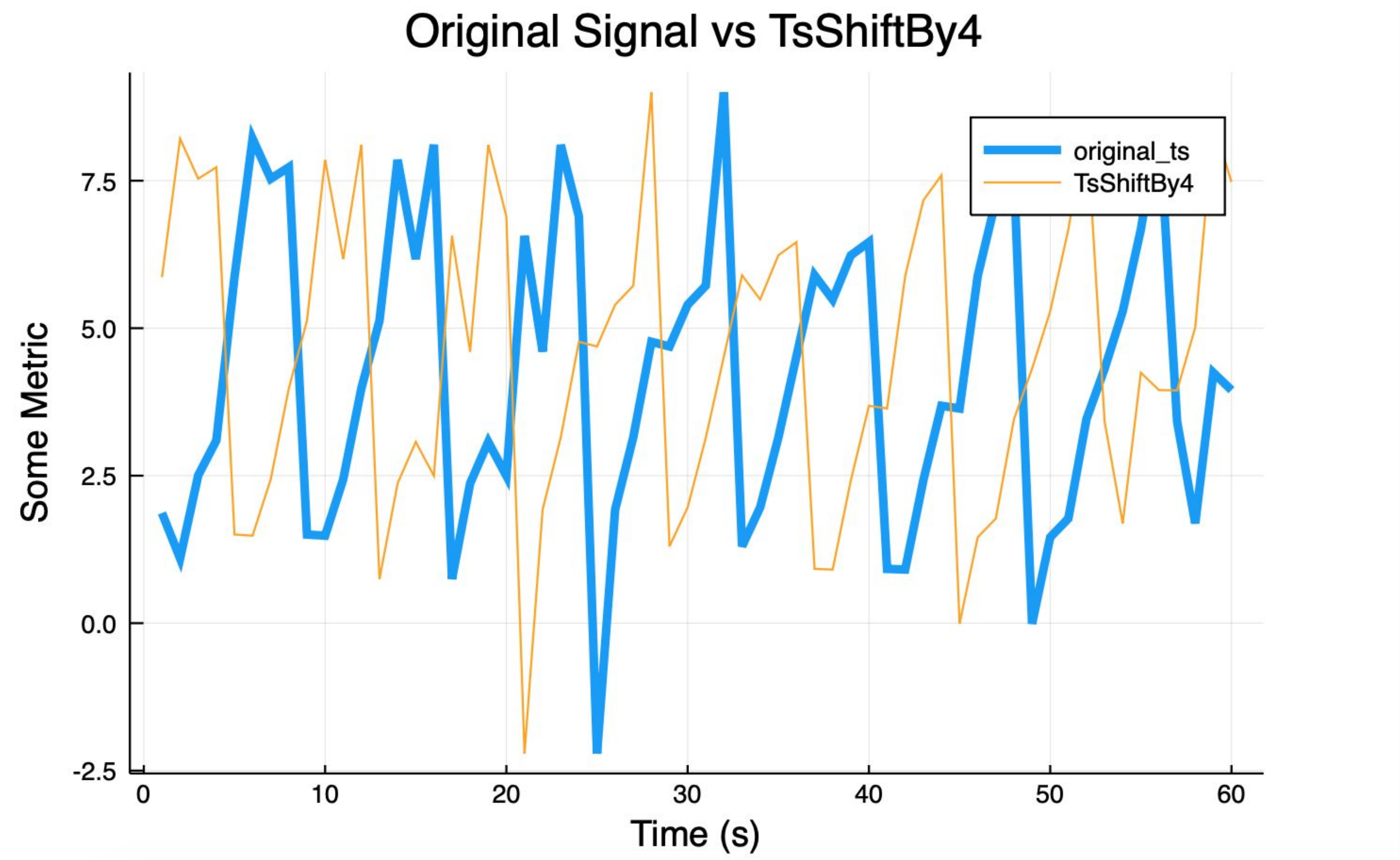
In [6]:
```
# Normalized cross-correlation can detect the correlation of two signals with different amplitudes:
# Notice we have perfect correlation between signal A and the same signal with half the amplitude!

proof = sum(a .* (a/2)) / sqrt(sum(a.^2) * sum((a./2).^2))
```
Out[6]: 1.0

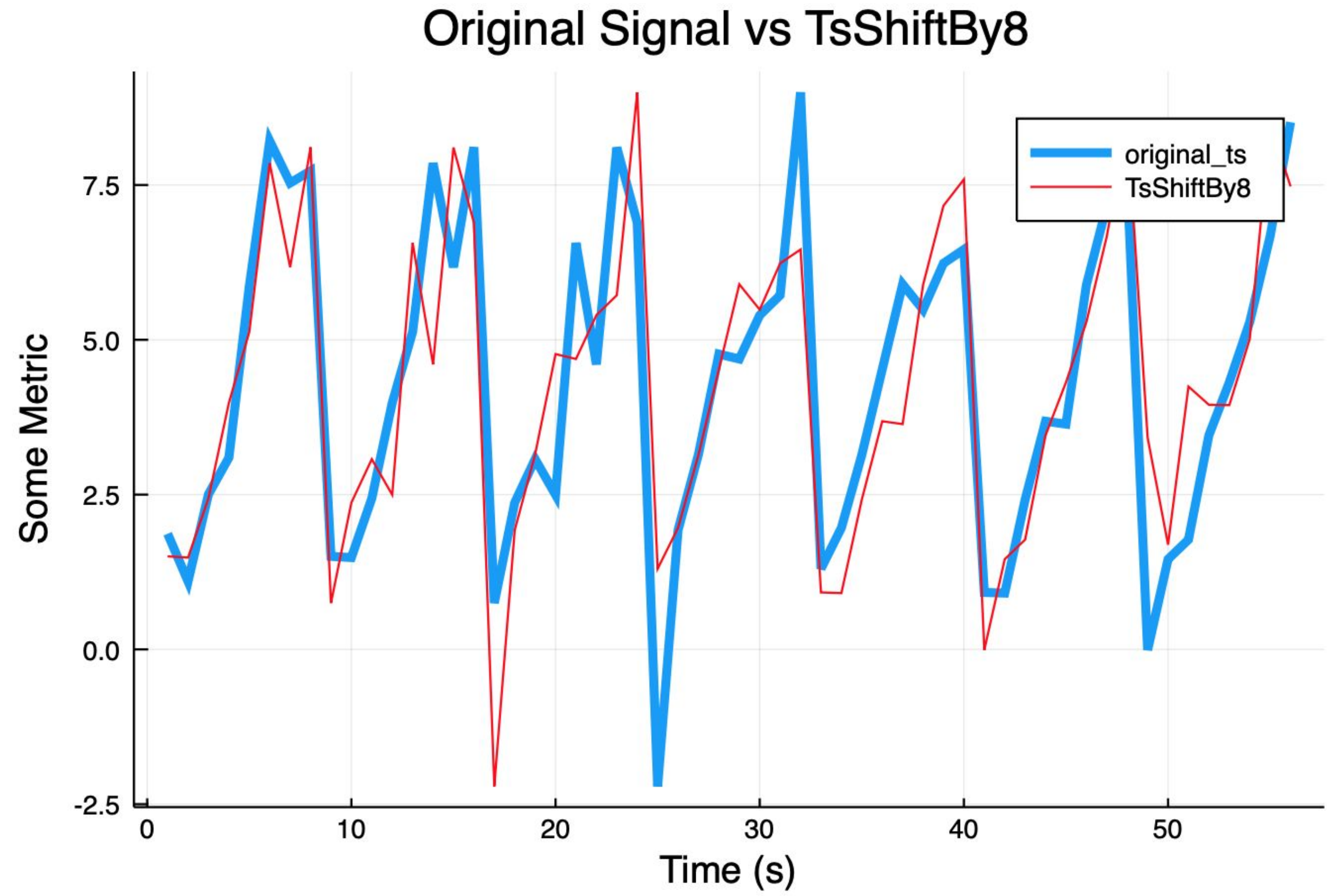
Autocorrelation:

One is the Loneliest Number!



Original Signal
Some Metric
7.5
5.0
2.5
0.0
-2.5
0
10
20
30
40
50
60
Time (s)
["original_ts"]


Original Signal vs TsShiftBy4
Some Metric
Time (s)
original_ts
TsShiftBy4
7.5
5.0
2.5
0.0
-2.5
0
10
20
30
40
50
60



Original Signal vs TsShiftBy8
Some Metric
Time (s)
original_ts
TsShiftBy8
7.5
5.0
2.5
0.0
-2.5
0
10
20
30
40
50

Autocorrelation

1. Comparison of time series with itself at different times
2. Auto-Correlation detects repeating patterns or “**seasonality**” !
3. Auto-Correlation answers questions like:
    a. Can we see some weekly pattern?
    b. Is today similar to last week today?

Unnormalized Autocorrelation

```
#Computing the Correlations -- here autocorrelations (i.e., multiplying and Summing the two

corr_shift4 = sum(autoSignalFour .* autoSignalShiftFour); # equals 948.4089186791925
corr_shift8 = sum(autoSignalEight .* autoSignalShiftEight); #equals 1336.069302482692l
```

Leads us to believe we have detected possible seasonality at 8

Normalized Autocorrelation Makes it Obvious

```
norm_auto_shift4 = sum(autoSignalFour .* autoSignalShiftFour) / sqrt(sum(autoSignalFour.^2)
norm_auto_shift8 = sum(autoSignalEight .* autoSignalShiftEight) / sqrt(sum(autoSignalEight.
```

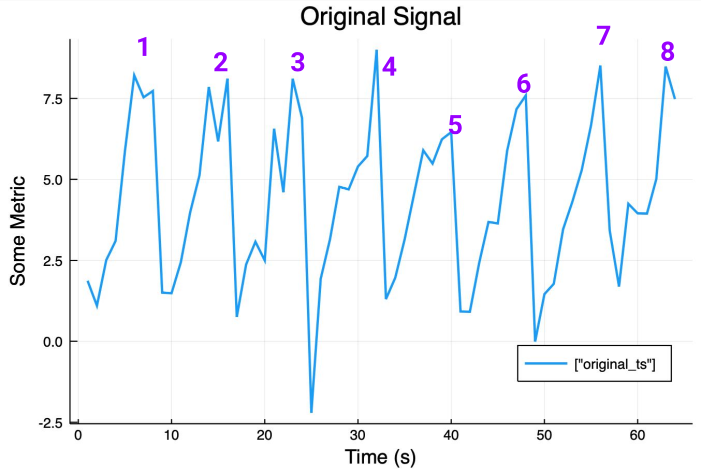
```
# norm_auto_shift4 = 0.6227933971623315
# norm_auto_shift8 = 0.9602671052926668  == Normalized autocorrelation makes it very obviou
```

In [12]:
```
rng = MersenneTwister(1234);
numRepeats = 8;
autoSignal = repeat(collect(1.0:8.0), numRepeats) + randn!(rng, zeros(numRepeats*8));
x = collect(1:length(autoSignal))
plot(x,autoSignal,
     title = "Original Signal",
     label =["original_ts"],
     xlabel="Time (s)",
     ylabel="Some Metric",
     legend=:bottomright,
     linecolor = 1,
     lw =2)
```

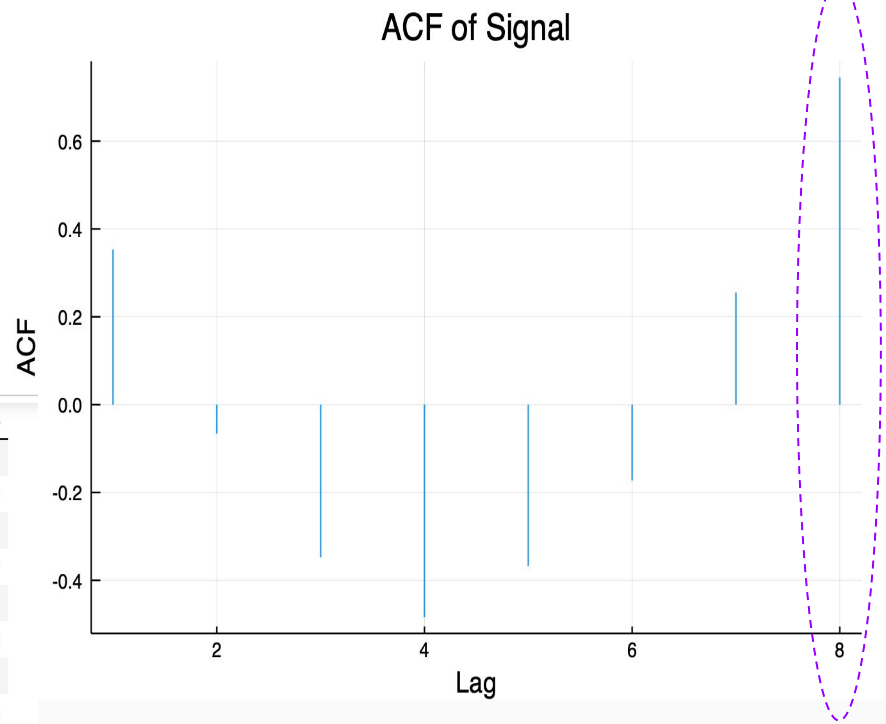
In [13]:
```
auto_r = autocor(autoSignal, collect(1:8));
```

15]:
```
plot(acf_df[!,:lag], acf_df[!,:acfValue], line = :sticks, legend=false,
    xlabel="Lag", ylabel="ACF", title = "ACF of Signal", )
```

15]:



	Int64	Float64
1	1	0.353117
2	2	-0.0659904
3	3	-0.347516
4	4	-0.48397
5	5	-0.36758
6	6	-0.1726
7	7	0.255374
8	8	0.744855



One More Bite at Time-Shifting:

Back to Cross-Correlations

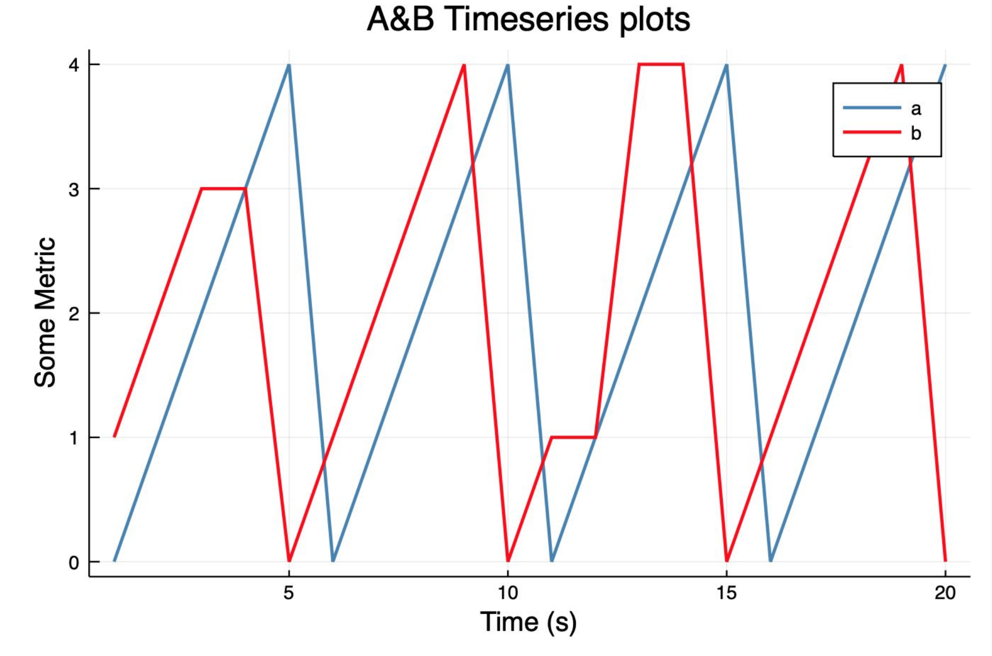
Cross-Correlation with Time Shifts

1. Check to see if one signal compared to another
    a. Lags (delays) - move elements to the right (t-lag)
    b. Leads (advancing) - move elements to left (t+lead)

2. Find the best time-shift at which time signals are correlated
    a. In autocorrelation, that is lag=lead=0 (most energy at perfect overlap)

In [16]:

```julia
a = [0,1,2,3,4,0,1,2,3,4,0,1,2,3,4,0,1,2,3,4]
b = [1,2,3,3,0,1,2,3,4,0,1,1,4,4,0,1,2,3,4,0]
time = collect(1:length(a));

plot(time, a, label="a", lw=2, linecolor=:steelblue, xlabel="Time (s)",
     ylabel="Some Metric", title = "A&B Timeseries plots")
plot!(time, b, label="b", lw=2, linecolor=:red)


```

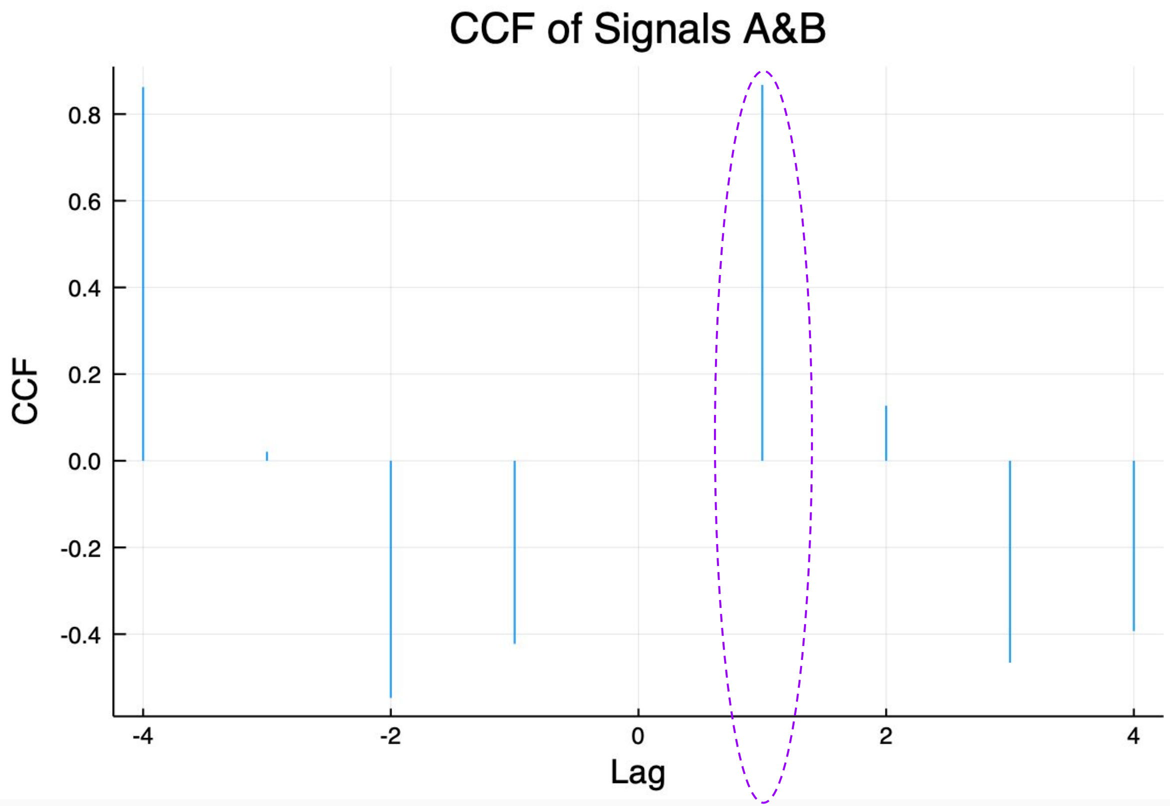


In [72]:

```
1
2  r = crosscor(b, a, collect(-4:4))
```

Out[21]:

	ccf_lag	ccfValue
	Int64	Float64
1	-4	0.86232
2	-3	0.0210021
3	-2	-0.547289
4	-1	-0.422512
5	0	-3.29181e-17
6	1	0.867262
7	2	0.127248
8	3	-0.465752
9	4	-0.392862



Lag = 1 => Best Correlation

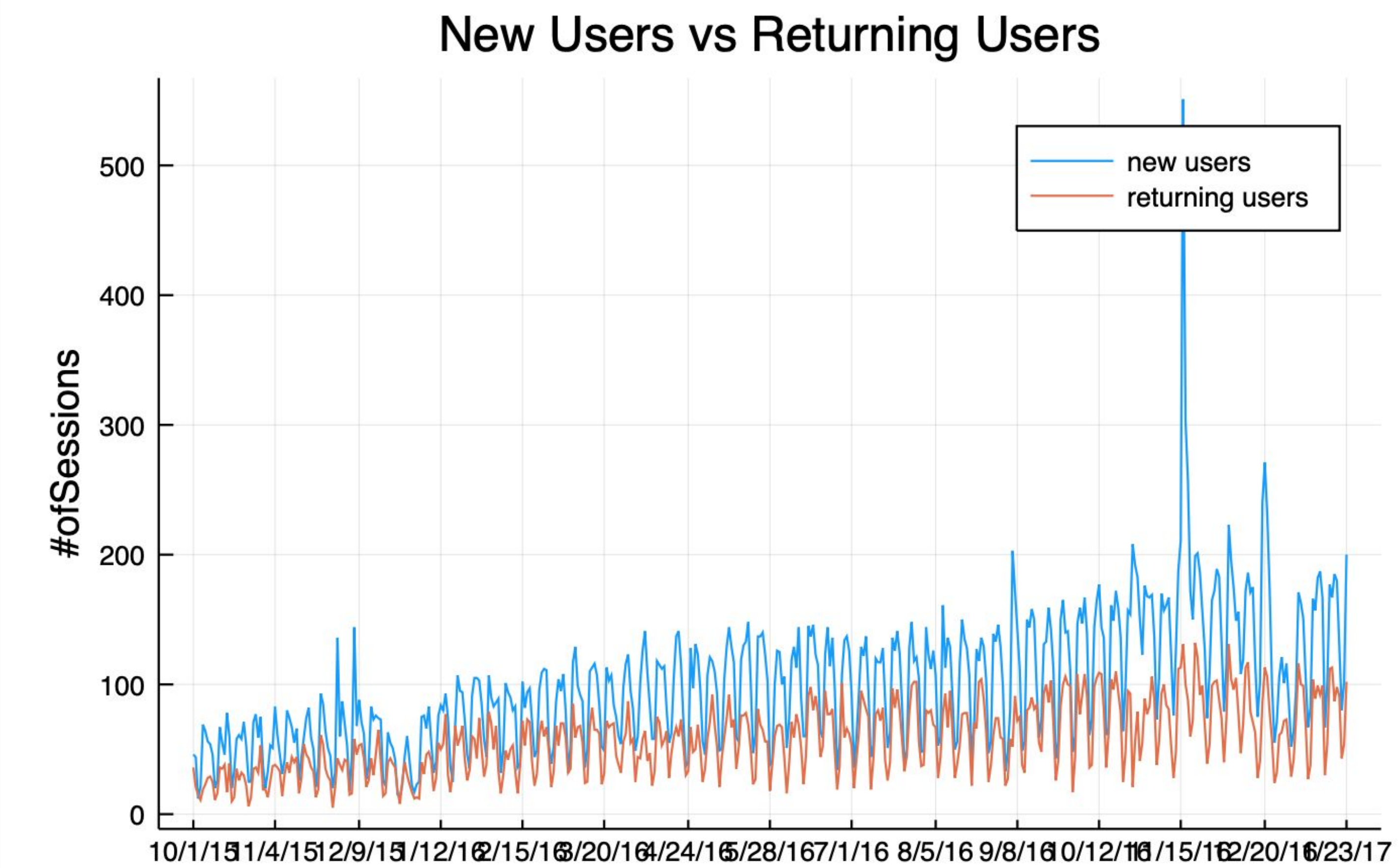
Okay Smarty,
How Does Correlation Help with Anomaly Detection?

Checkout: Key Performance Indicators (KPIs)

https://anomaly.io/detect-correlation-time-series/index.html

New Users vs Returning Users

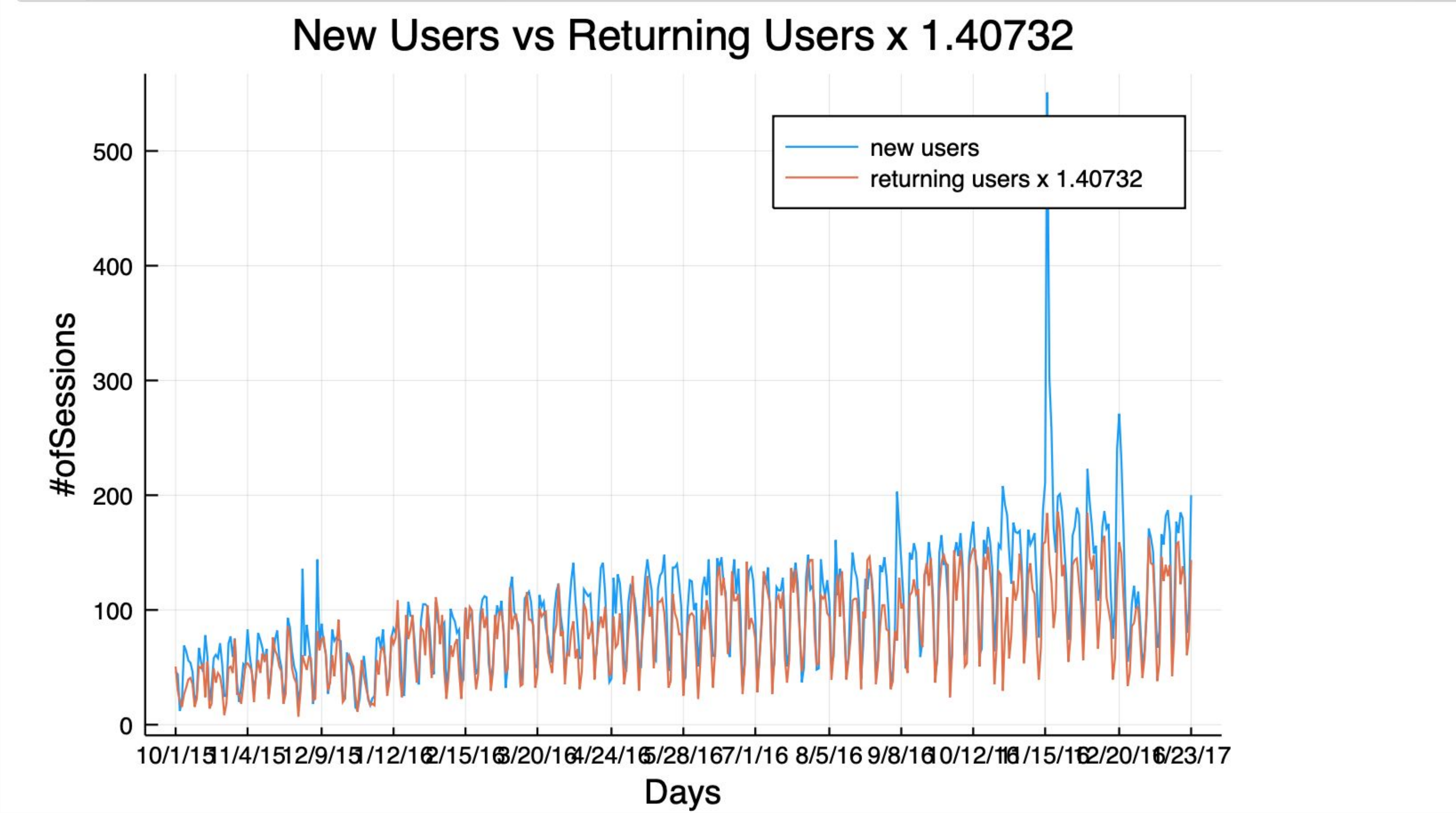
to a website



In [27]:
```
crosscor(new_data_df[!,:Sessions], return_data_df[!,:Sessions], [0])
```

Out[27]:
```
1-element Array{Float64,1}:
 0.8371268696198125
```

Show high correlation > 0.7



In [28]:

```
# Subtract the time Series
multi = sum(new_data_df[!,:Sessions]/return_data_df[!,:Sessions])/ length(return_data_df[!,:Sessions])
multi #display 1.40732
align_return_data = return_data_df[!,:Sessions].*multi;
```

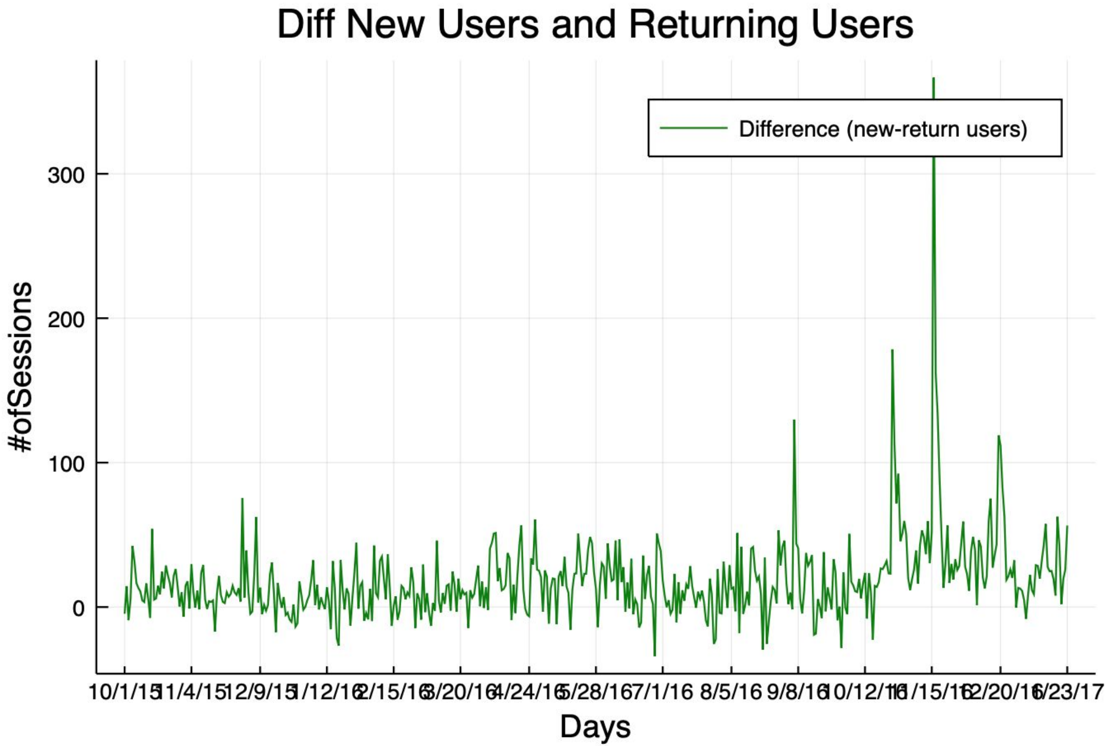
In [31]:

```julia
subtractTs = new_data_df[!,:Sessions] - align_return_data;
```

In [32]:

```julia
plot(return_data_df[!,:Day_Index], subtractTs,
     xlabel="Days", ylabel="#ofSessions", title = "Diff New Users and Returning Users",
     label="Difference (new-return users)", linecolor=:green)
```

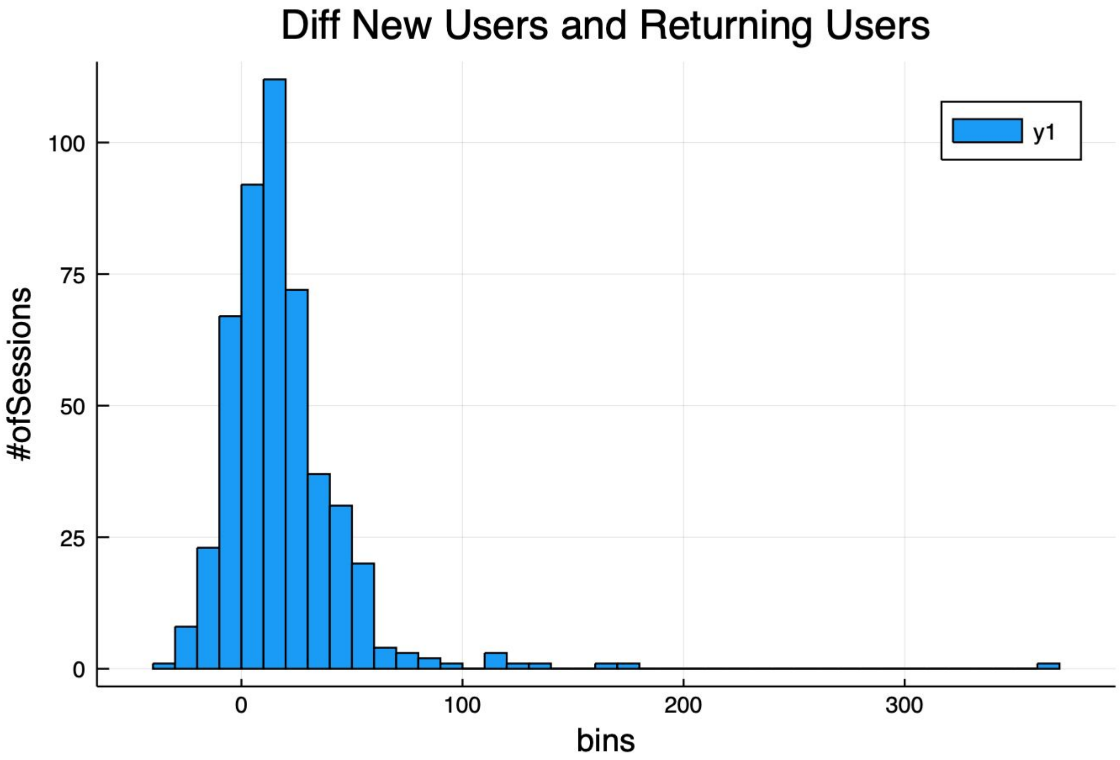
Out[32]:



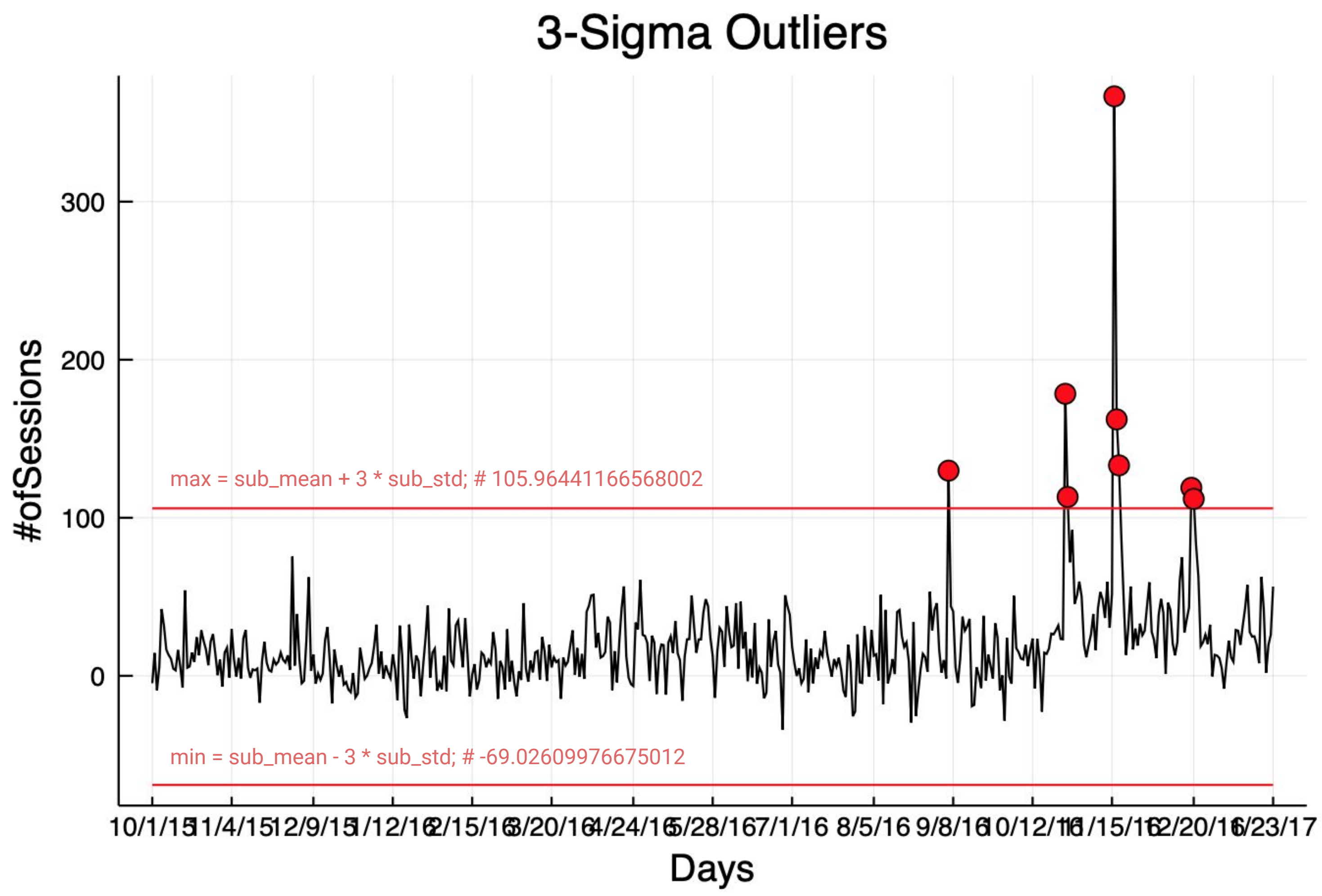
In [33]:

```
# Finding Outliers in Correlated Time Series
histogram(subtractTs, xlabel="bins", ylabel="#ofSessions", title = "Diff New Users and Returning Users
```

Out[33]:



Out[38]:


3-Sigma Outliers
#ofSessions
300
200
100
0
max = sub_mean + 3 * sub_std; # 105.96441166568002
min = sub_mean - 3 * sub_std; # -69.02609976675012
10/1/15
11/4/15
12/9/15
1/12/16
2/15/16
3/20/16
4/24/16
5/28/16
7/1/16
8/5/16
9/8/16
10/12/16
11/15/16
12/20/16
1/23/17
Days

	index	value	date
	Int64	Float64	String
1	342	129.819	9/6/16
2	392	178.446	10/26/16
3	393	113.19	10/27/16
4	413	366.641	11/16/16
5	414	162.268	11/17/16
6	415	133.155	11/18/16
7	446	118.97	12/19/16
8	447	111.972	12/20/16