Anomaly Detection using Statistical Learning for Identifying Possible Heart Attacks

Group 4

Jiachen Hu, Jiayi Wang, Marcus Loke, Qinyun Luo, Rohit Kundurthi, Xin Li, Yafei Wen, Yinjie Dai, Yunzi Ma, Zheyu Xu

Please visit our GitHub repository for the full report and codes:
https://github.com/markielokie/anomaly-detection

Introduction

What is anomaly detection?

A form of data analysis that identifies outliers / exceptional occurrences that deviate significantly and materially from normal / expected behavior.

Why is it important for business and society?

Can help predict and prevent unfavorable outcomes across different circumstances (e.g., flag heart attacks immediately, trigger stop-losses in trading, trust and safety issues)

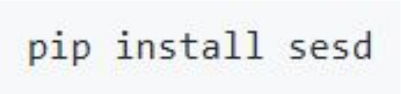
Why are we going to discuss today?

Overview of package installation and functions included, explanation of application to a real-world example, additional applications and conclusion

Installation and Importing of Package

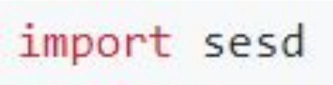
- **Name of Package**
  - **Sesd** - seasonal esd/esd

- **Method of installation**
  - Using **pip** function to install the package of sesd



```
pip install sesd
```

- **Importing Package**



```
import sesd
```

Explanation:
Function Selection

*We will use **seasonal_esd()** for the practical example because it can <u>filter the trend and seasonal components</u> of the heartrate dataset and use <u>robust statistical metrics</u> to improve accuracy.*

esd

(df_ts, max_anomalies=5, alpha=0.05, hybrid=True)

Computes the **Extreme Studentized Deviate** of a time series (*assumes a normal data distribution)*.

A **Grubbs Test** is performed *max_anomalies* times with the caveat that each time the top value is removed.

$$C = \frac{\max_t \mid x_t - \overline{x} \mid}{s}$$

If **hybrid** [high percentage (up to 50%) of anomalies] :

calculate with **median and Median Absolute Deviation**

If **not hybrid**:

calculate with **mean and standard deviation**

seasonal_esd

(df_ts, periodicity=1440, hybrid=True, max_anomalies=5, alpha=0.05)

Computes the **Seasonal Extreme Studentized Deviate** of a time series.

Steps:

- Decompose the time series into **STL decomposition**
- If **hybrid**:
  - calculate the **MAD**

  If **not hybrid**:
  - calculate the **median**
- Perform a regular **ESD test** on the **residual**:

  R = ts - seasonality - MAD or median

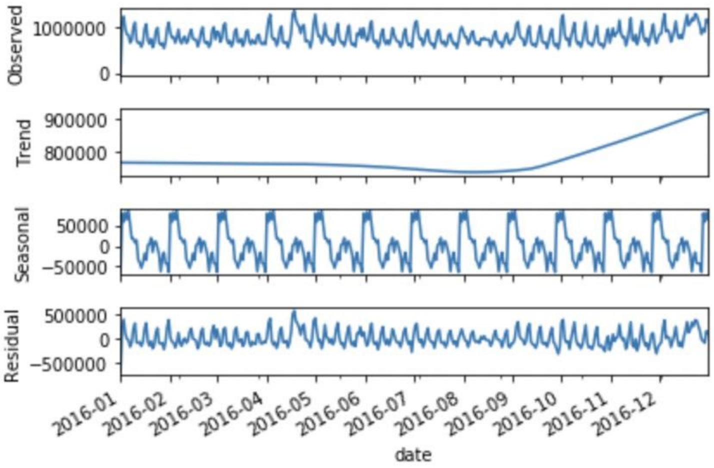
Returns: The indices of the anomalies in the time series.

Explanation

STL decomposition

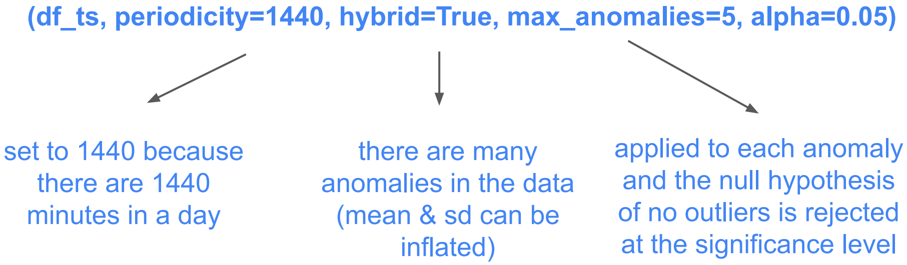
Seasonal and Trend decomposition using Loess

This is a method of using locally fitted regression model to decompose a Time Series data into 3 components containing **seasonality, trend and residual**. It always used in economics and environmental analysis.



Parameters & Arguments

ts: The time series to compute the SESD.
periodicity: The statsmodel library requires a periodicity to compute the STL decomposition.
hybrid: False – mean and standard deviation; True – median and MAD (calculate the z-scores).
max_anomalies: The number of times the Grubbs' Test will be applied to the time series.
alpha: the significance level.



(df_ts, periodicity=1440, hybrid=True, max_anomalies=5, alpha=0.05)

set to 1440 because there are 1440 minutes in a day

there are many anomalies in the data (mean & sd can be inflated)

applied to each anomaly and the null hypothesis of no outliers is rejected at the significance level

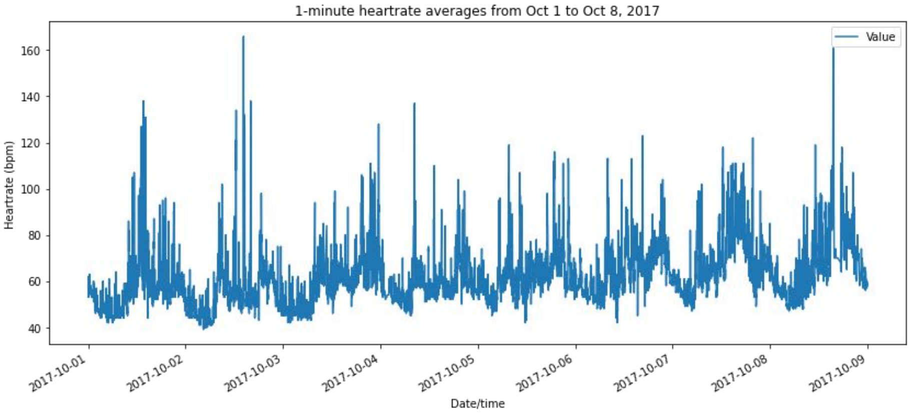
Identifying Possible Heart Attacks (Demo)

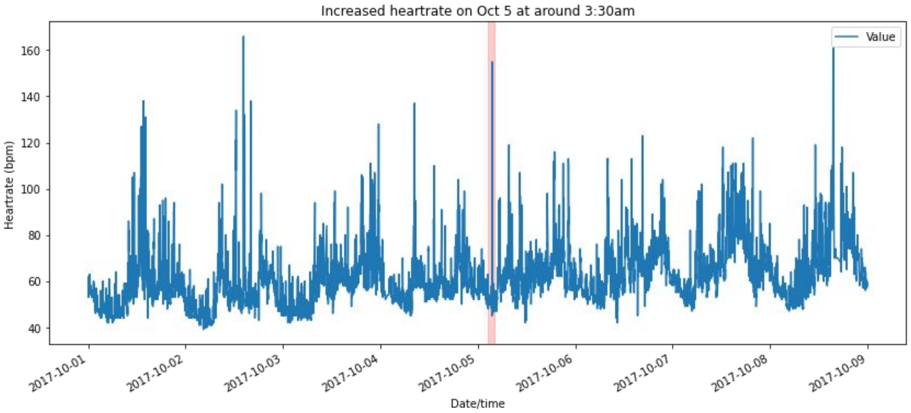
Heartrate data from *Fitabase*

- 1-minute heartrate averages from Oct 1 to Oct 8, 2017
- 8-days worth of data
- Obvious spikes in midday/afternoon

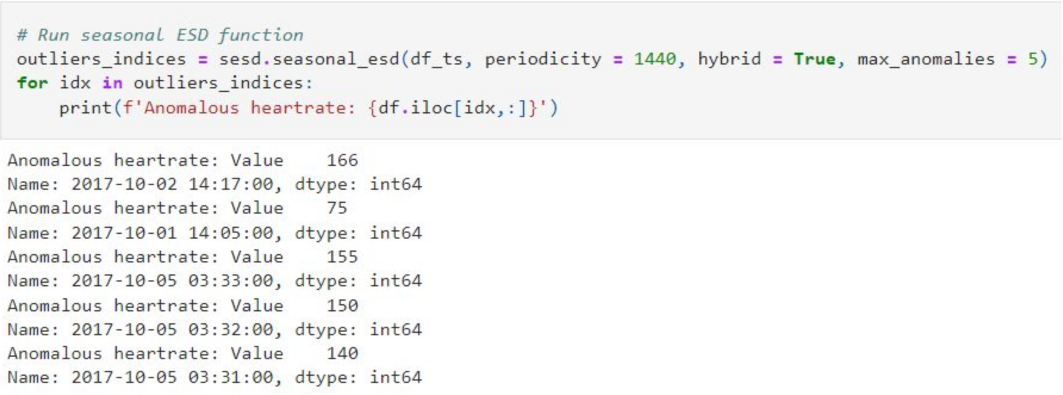
Insert artificial anomalies

- Insert increased heartrate (140~155 bpm) at 3:30am (in red)
- Normal human beings should be sleeping and not have such heartrate



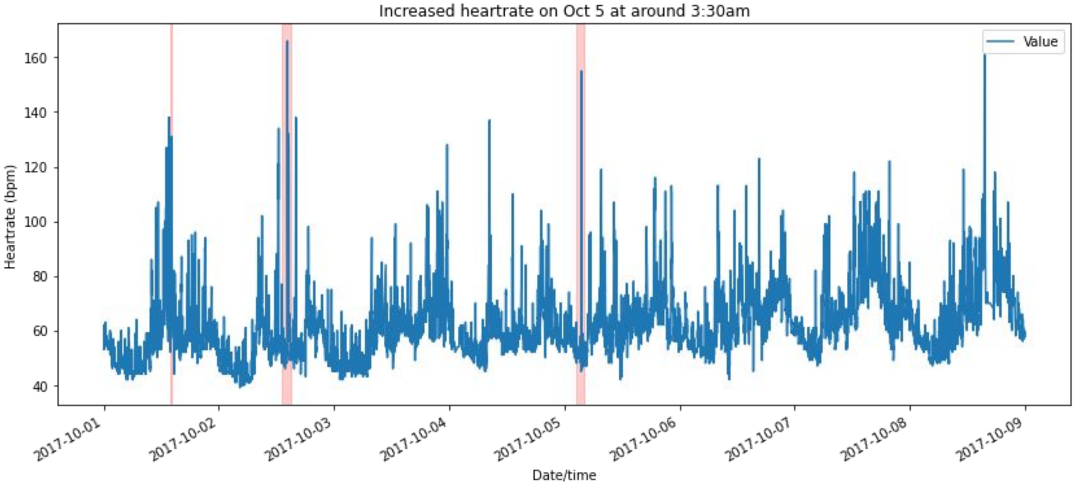


Identifying Possible Heart Attacks (Demo)



```
# Run seasonal ESD function
outliers_indices = sesd.seasonal_esd(df_ts, periodicity = 1440, hybrid = True, max_anomalies = 5)
for idx in outliers_indices:
    print(f'Anomalous heartrate: {df.iloc[idx,:]}')
```

```
Anomalous heartrate: Value    166
Name: 2017-10-02 14:17:00, dtype: int64
Anomalous heartrate: Value    75
Name: 2017-10-01 14:05:00, dtype: int64
Anomalous heartrate: Value    155
Name: 2017-10-05 03:33:00, dtype: int64
Anomalous heartrate: Value    150
Name: 2017-10-05 03:32:00, dtype: int64
Anomalous heartrate: Value    140
Name: 2017-10-05 03:31:00, dtype: int64
```



Increased heartrate on Oct 5 at around 3:30am

Results

- Anomaly detection algorithm has successfully flagged the three artificial anomalies inserted at around 3:30am on Oct 5 (heartrates of 140, 150 and 155 bpm respectively).

- An anomalous heartrate of 166 bpm was also detected on Oct 2, 2:17pm, which makes sense because the heartrate was at its peak over this 8-day period.

- Interestingly, an anomalous heartrate of 75 bpm was flagged on Oct 1, 2:05pm, which is pretty strange because the heartrate does not seem that high. A possible explanation could be that this is a local anomaly that would otherwise be masked by seasonal data.

Conclusion

For the purpose of better detecting the anomalies of human heart rate and preventing heart attack, our team applied anomaly detection (seasonal ESD) function to achieve that. The anomaly detection algorithm not only successfully identifies all the artificial anomalies, but also an recorded anomalous heart rate of 166 bpm. Interestingly, a heart rate anomaly of 75 bpm was flagged out.This may be a local anomaly.

- Other applications of anomaly detection:

monitoring instrument to detect rate of heart to ensure patient health

Self-monitoring smart to monitor users' health

veterinary equipment to cure animals

Other wave anomaly detection

References

https://github.com/nachonavarro/seasonal-esd-anomaly-detection

https://arxiv.org/pdf/1704.07706.pdf