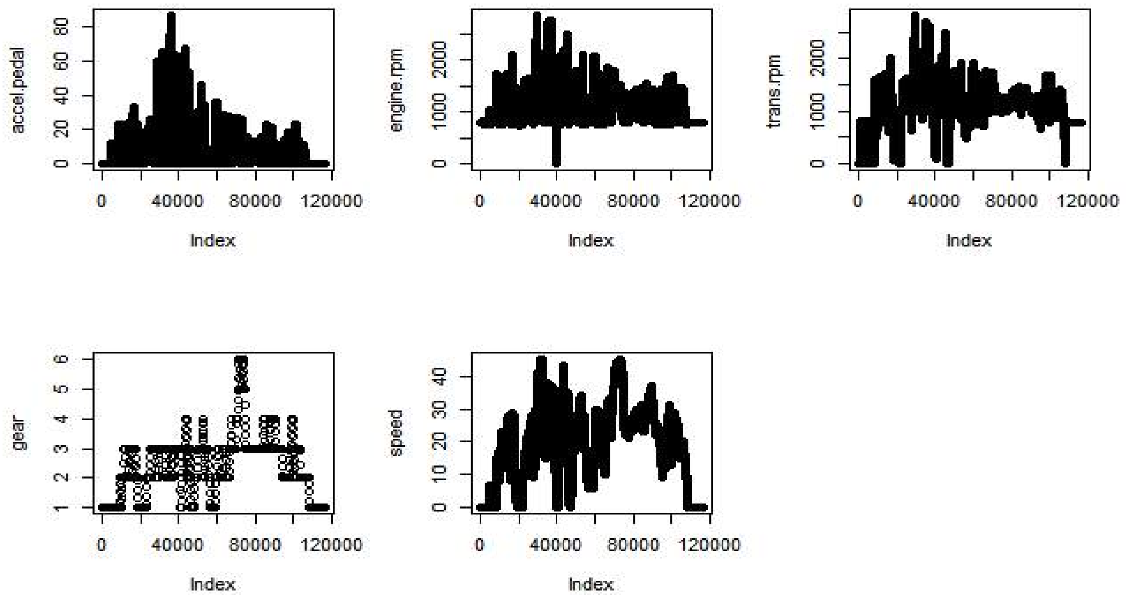
Sensor Data Analysis

방송대 백재욱, Univ. of Minnesota-Morris 김종민, 중앙대 김대경

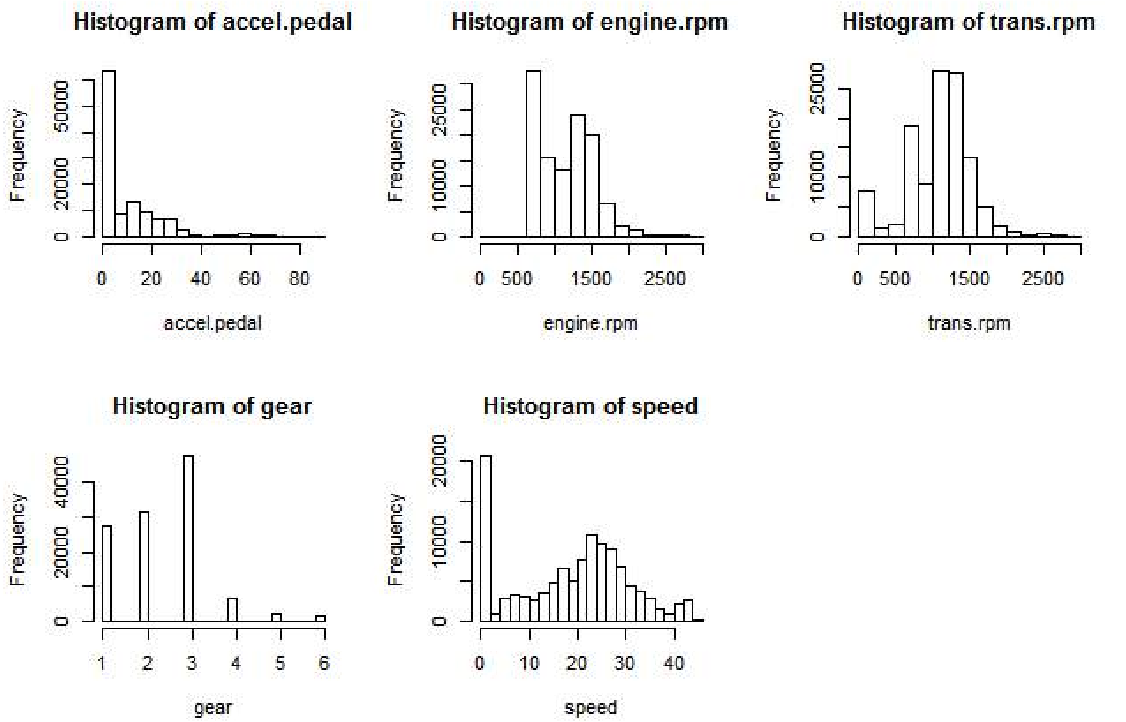
1. Exploratory Data Analysis

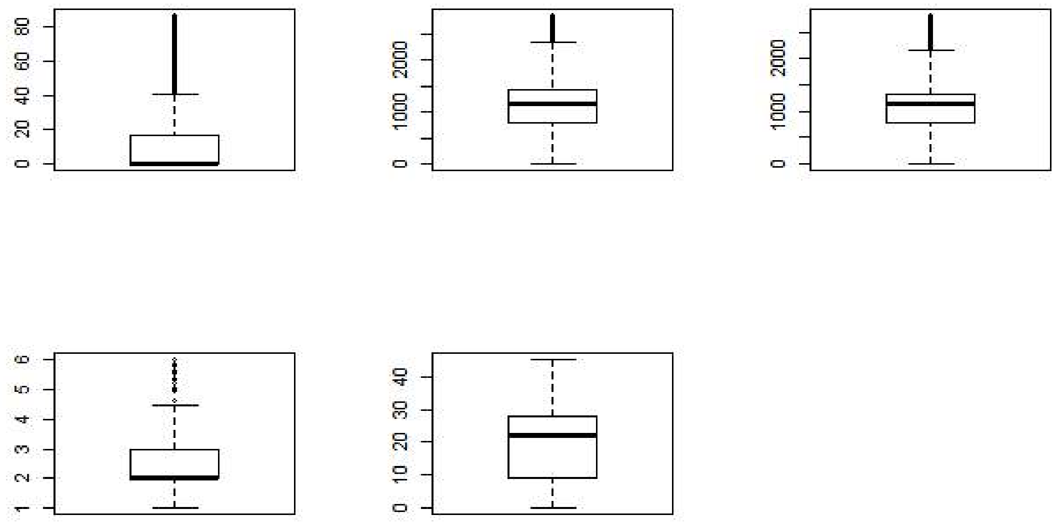
```
head(signal1)
  time accel.pedal engine.rpm trans.rpm gear speed
1    0           0   788.0000         0    1     0
2    1           0   788.0000         0    1     0
3    2           0   787.9296         0    1     0
4    3           0   787.2473         0    1     0
5    4           0   787.0000         0    1     0
6    5           0   787.0000         0    1     0
0.02*nrow(signal1)
[1] 2331.86   # total of 2331.86 seconds
(0.02*nrow(signal1))/60
[1] 38.86433 # total of 38.86433 minutes
```



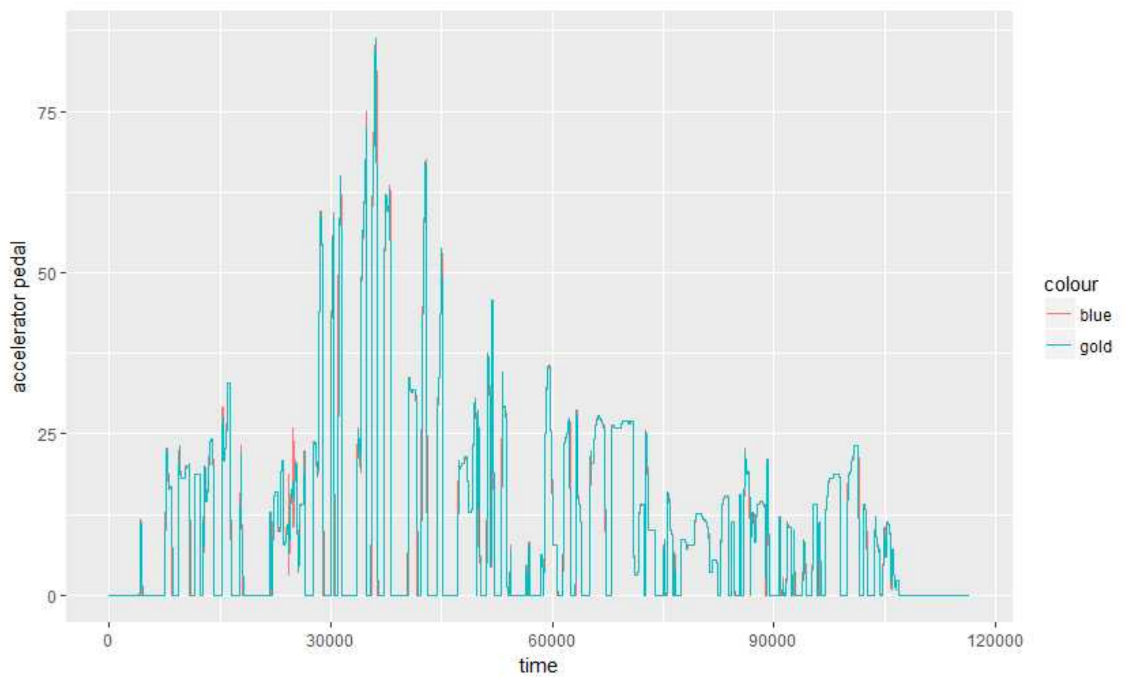
summary(signal1)

```
      time          accel.pedal       engine.rpm        trans.rpm           gear              speed
 Min.   :     0   Min.   : 0.000   Min.   :   0.0   Min.   :   0.0   Min.   :1.000   Min.   : 0.00
 1st Qu.: 29148   1st Qu.: 0.000   1st Qu.: 792.9   1st Qu.: 786.8   1st Qu.:2.000   1st Qu.: 9.00
 Median : 58296   Median : 0.000   Median :1156.3   Median :1145.6   Median :2.000   Median :22.00
 Mean   : 58296   Mean   : 9.783   Mean   :1165.4   Mean   :1086.2   Mean   :2.401   Mean   :19.26
 3rd Qu.: 87444   3rd Qu.:16.351   3rd Qu.:1416.6   3rd Qu.:1334.8   3rd Qu.:3.000   3rd Qu.:28.00
 Max.   :116592   Max.   :86.275   Max.   :2846.4   Max.   :2812.9   Max.   :6.000   Max.   :45.00
```





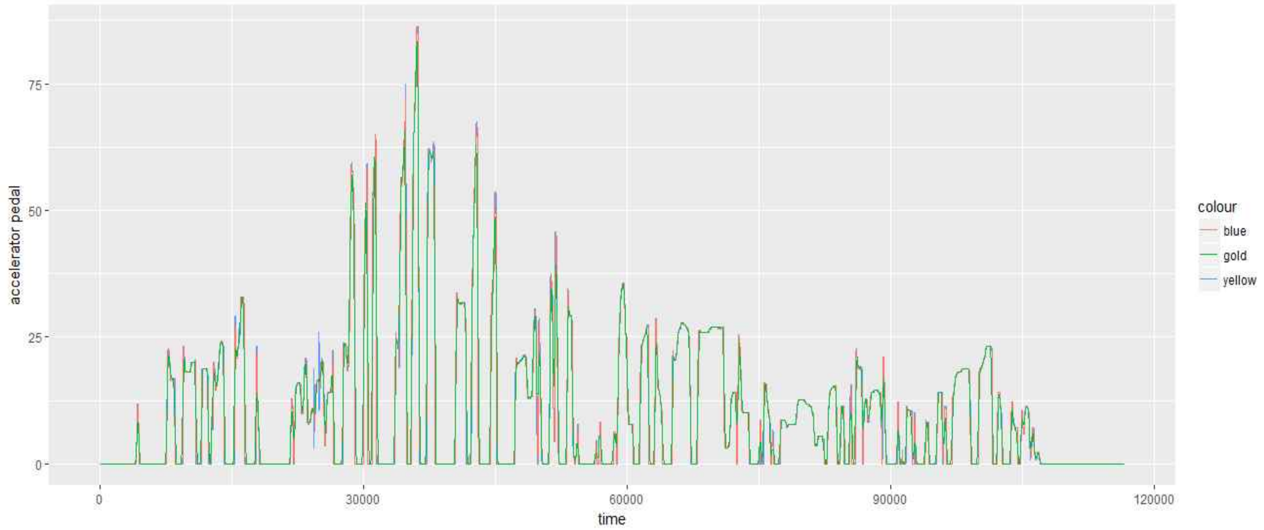
```
ggplot() + geom_line(data = signal1, aes(x = time, y = accel.pedal, colour = "blue"))
+ geom_line(data = df50, aes(x = time50, y = mean50.accel, colour = "gold")) +
ylab('accelerator pedal')
# single, 50 frames together
```




```
ggplot() + geom_line(data = signal1, aes(x = time, y = accel.pedal, colour = "yellow"))
+ geom_line(data = df50, aes(x = time50, y = mean50.accel, colour = "blue")) +
geom_line(data = df300, aes(x = time300, y = runMean300, colour = "gold")) +
```

ylab('accelerator pedal')

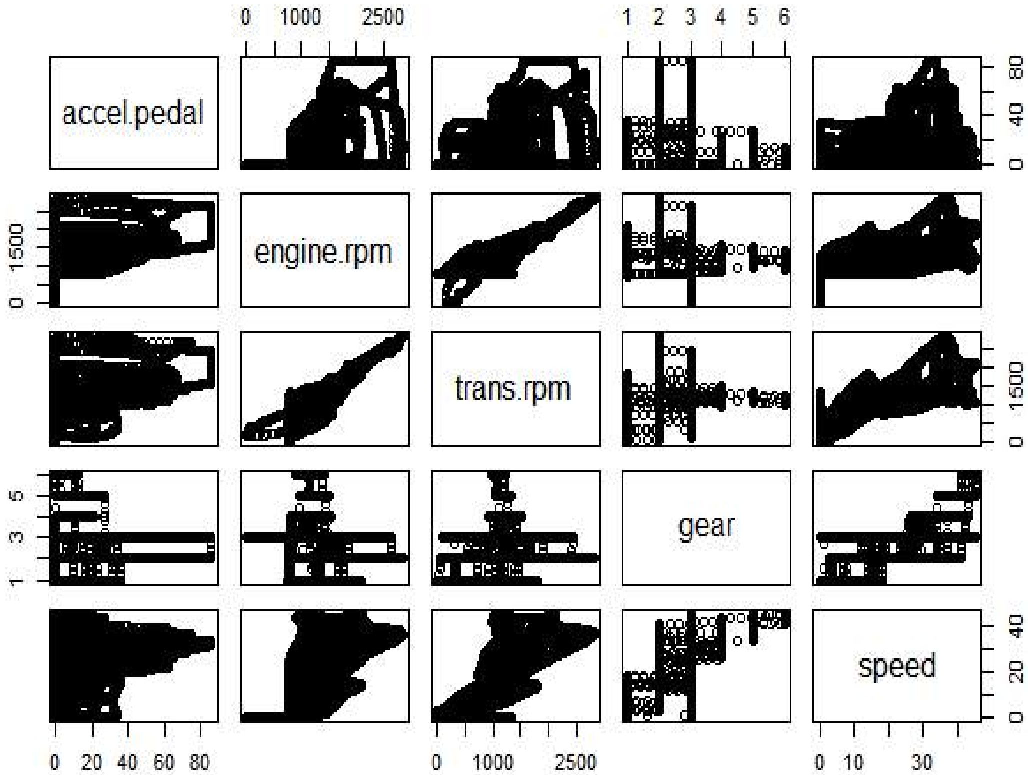
single, 50 frame, 300 frame together



cor(signal1)

	accel.pedal	engine.rpm	trans.rpm	gear	speed
accel.pedal	1.00000000	**0.6**476683	0.4638068	0.09788245	0.2675681
engine.rpm	0.64766830	1.0000000	**0.8**195003	0.18938163	**0.5**581749
trans.rpm	0.46380682	0.8195003	1.0000000	0.40580195	**0.7**342482
gear	0.09788245	0.1893816	0.4058019	1.00000000	**0.8**537444
speed	0.26756814	0.5581749	0.7342482	0.85374437	1.0000000

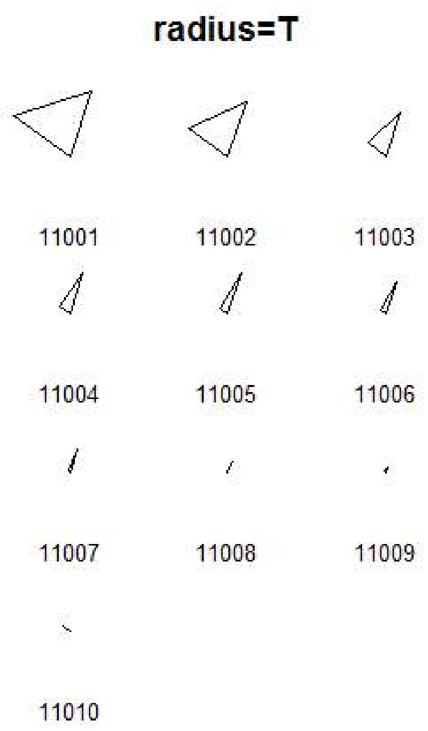
plot(signal1)



Let X1, X2, X3, X4, X5 be accel.pedal, engine.rpm, trans.rpm, gear and speed.
1.1 correlation between X1 at t and X2 at t+j, j=1, 2, ..., 50 (50==1sec)
HW1:
1.1 correlation between X1 at t and X2 at t+j, j=1, 2, ..., 50 (50==1sec)
1.2 correlation between X2 at t and X3 at t+j, j=1, 2, ..., 50 (50==1sec)
1.3 correlation between X2 at t and X5 at t+j, j=1, 2, ..., 50 (50==1sec)
1.4 correlation between X3 at t and X5 at t+j, j=1, 2, ..., 50 (50==1sec)
1.5 correlation between X4 at t and X5 at t+j, j=1, 2, ..., 50 (50==1sec)

[Q1: Reaction time from acceleration pedal to engine, transmission, gear and speed?]

```
stars(s.signal1,radius="T",main="radius=T")
```



2. Control Charts

2.1 X control chart

2.1.1 X control chart for accel.pedal

```
accel.pedal.chart=qcc(accel.pedal,type="xbar.one", title="X관리도 for accel.pdeal", xlab="time", ylab="accel.pedal")
summary(accel.pedal.chart)
```

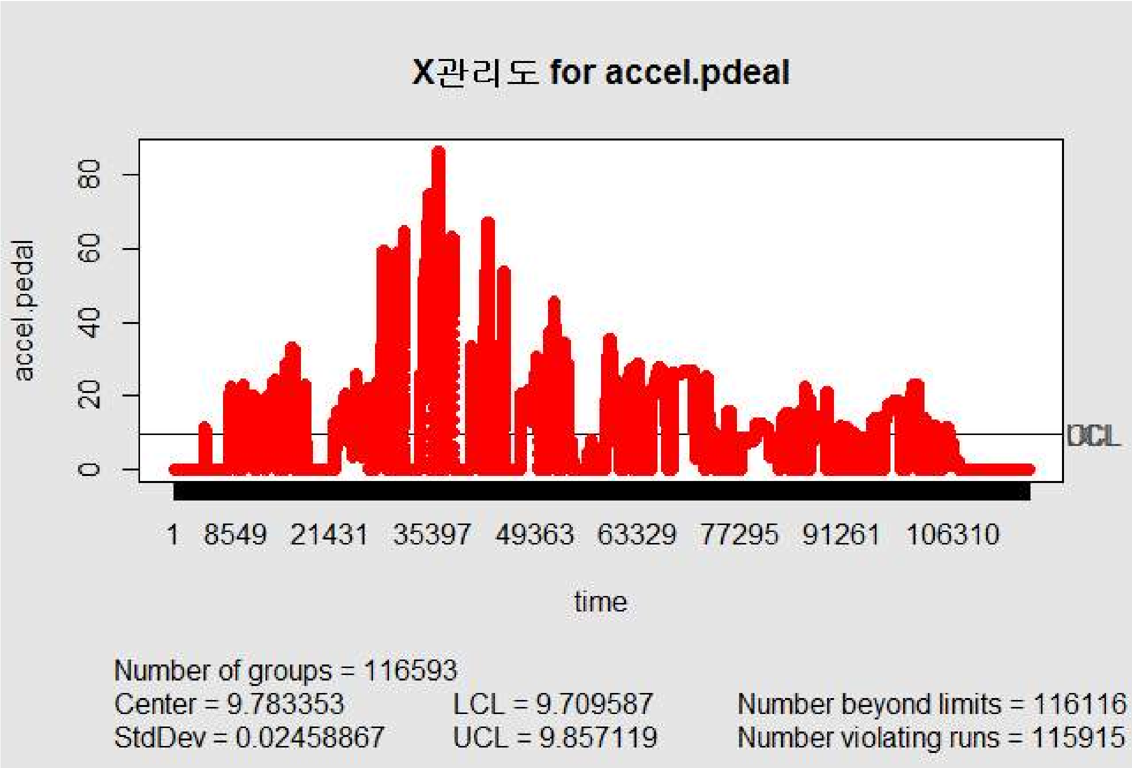
```
Call:
qcc(data = accel.pedal, type = "xbar.one", title = "X관리도 for accel.pdeal",     xlab =
"time", ylab = "accel.pedal")

xbar.one chart for accel.pedal

Summary of group statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   9.783  16.350  86.270

Group sample size:  1
Number of groups:  116593
Center of group statistics:  9.783353
Standard deviation:  0.02458867

Control limits:
      LCL      UCL
 9.709587 9.857119
```



2.1.2 X control chart for engine.rpm

2.1.3 X control chart for trans.pedal

2.1.4 X control chart for gear

2.1.5 X control chart for speed

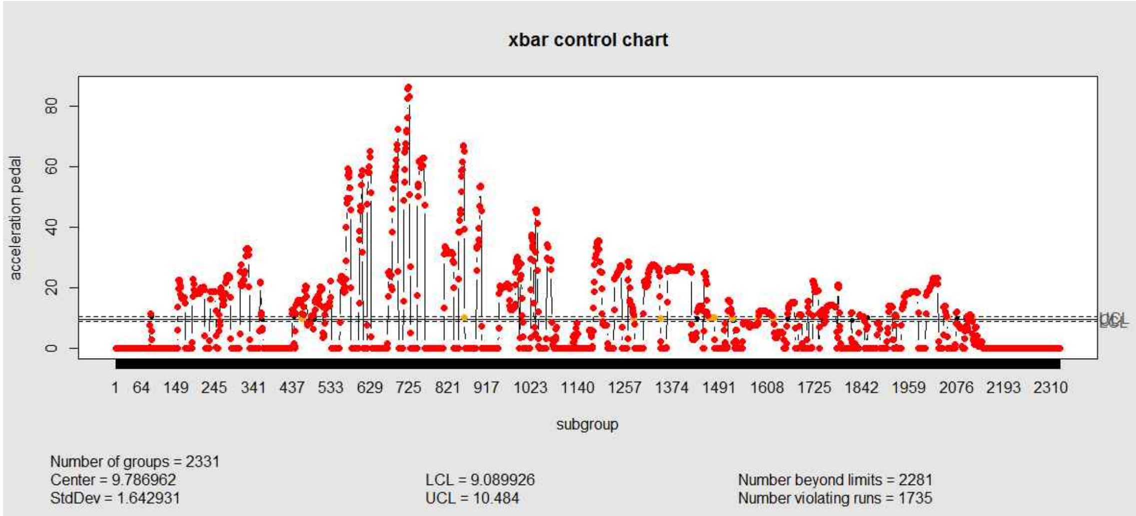
2.2 MR control chart

2.3 xbar control chart

```
dim(accel.data)
```
[1] 2331 50

2.3.1 xbar control chart for accel.pedal

```
accel.xbar.chart=qcc(accel.data,type="xbar",title="xbar control chart", xlab="subgroup", ylab="acceleration pedal")
```



summary(accel.xbar.chart)

Call:
qcc(data = accel.data, type = "xbar", title = "xbar control chart", xlab = "subgroup", ylab = "acceleration pedal")

xbar chart for accel.data

Summary of group statistics:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.000	0.000	1.698	9.787	16.140	86.270

Group sample size: 50
Number of groups: 2331
Center of group statistics: 9.786962
Standard deviation: 1.642931

Control limits:

LCL	UCL
9.089926	10.484

2.3.2 xbar control chart for X2

2.3.3 xbar control chart for X3

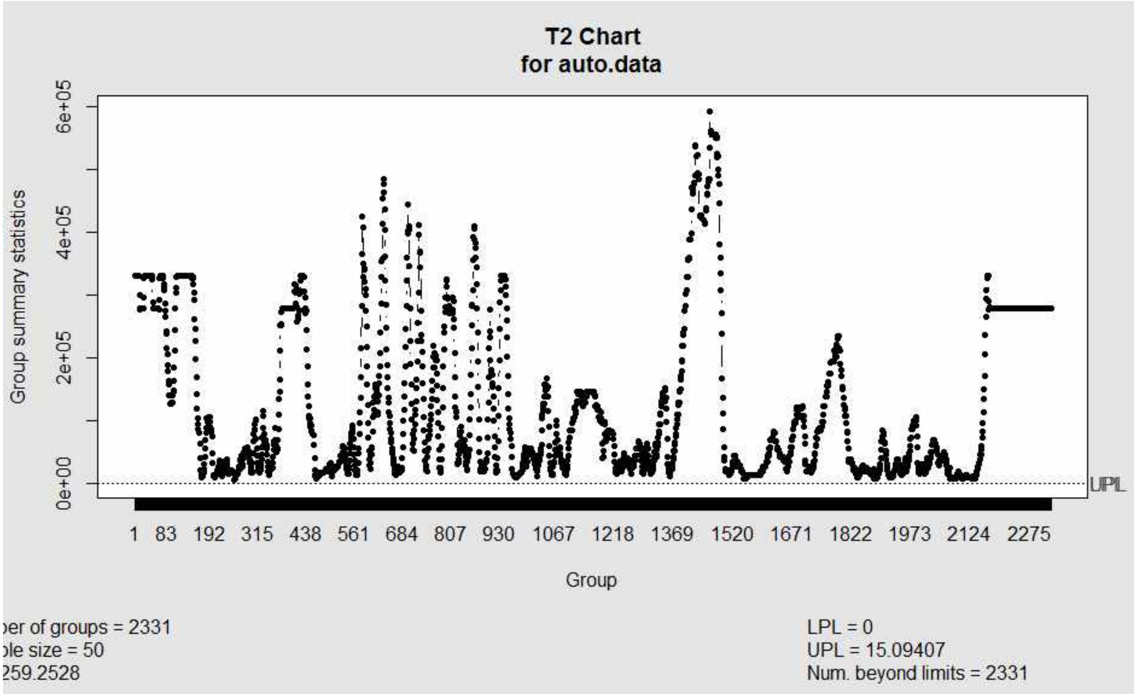
2.4 R control chart

2.4.1 R control chart for X1

2.4.2 R control chart for X2

2.5 Multivariate control chart

mcc = mqcc(auto.data, type = "T2", limits=FALSE, pred.limits=TRUE, confidence.level = 0.99)




```
summary(mcc)
Call:
mqcc(data = auto.data, type = "T2", limits = FALSE, pred.limits = TRUE,       confidence.level
= 0.99)

T2 chart for auto.data

Summary of group statistics:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   6213   25770   68850  127700  233800  593500

Number of variables:  5
Number of groups:  2331
Group sample size:  50

Center:
auto.data[1] auto.data[2] auto.data[3] auto.data[4] auto.data[5]
    9.786962  1165.490830  1086.349272     2.401109    19.265699

Covariance matrix:
               auto.data[1] auto.data[2] auto.data[3]  auto.data[4] auto.data[5]
auto.data[1]  2.6992105282    6.9941668  -0.36894817 -0.0009920977 -0.058965892
auto.data[2]  6.9941667745  761.3567143 325.25781331 -0.1175203997   0.556736108
auto.data[3] -0.3689481685  325.2578133 543.59122625  0.0330449366   1.373983583
auto.data[4] -0.0009920977   -0.1175204   0.03304494  0.0046765580   0.001060438
auto.data[5] -0.0589658916    0.5567361   1.37398358  0.0010604382   0.074478026
|S|:  259.2528

Prediction limits:
 LPL       UPL
   0  15.09407
```

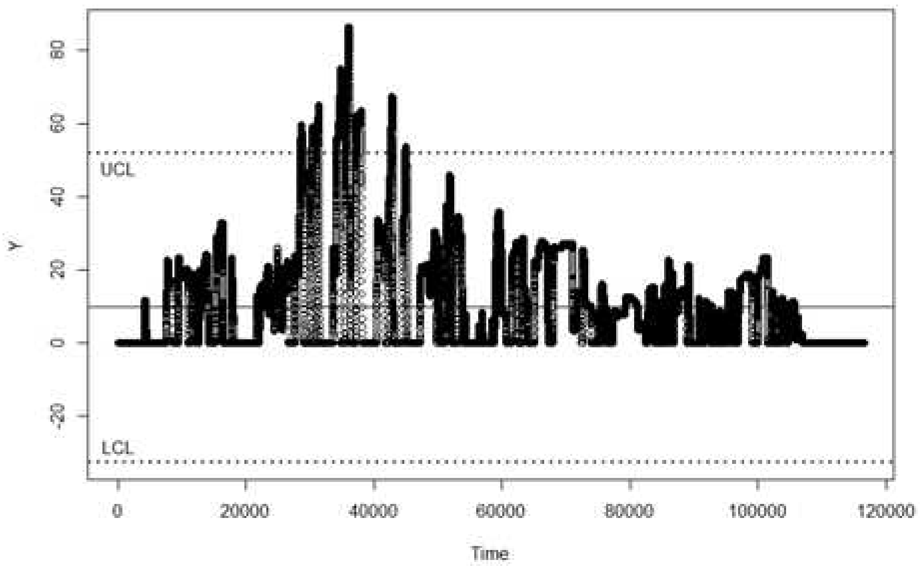
2.6 Copula-Based Time Series Model for Quality Control

#Joe.Markov.MLE Maximum Likelihood Estimation and Statistical Process Control Under the JoeCopula

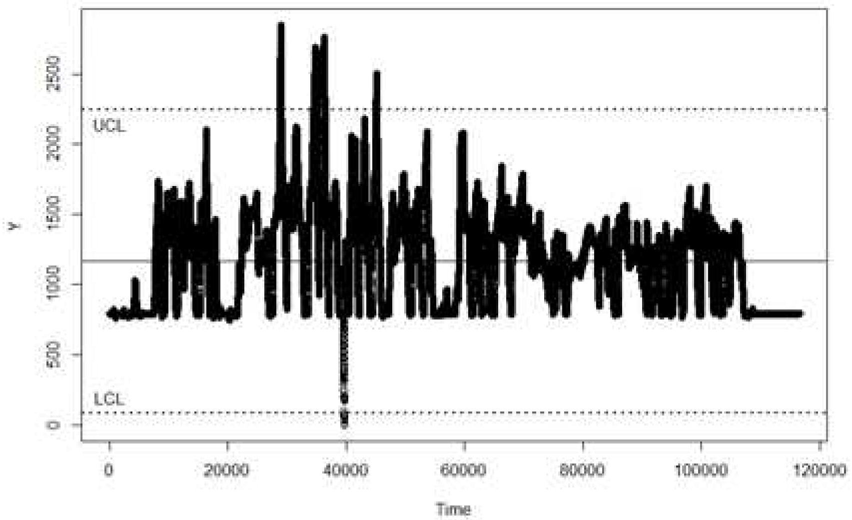
```
install.packages("Copula.Markov")
```

```
library(Copula.Markov)
```

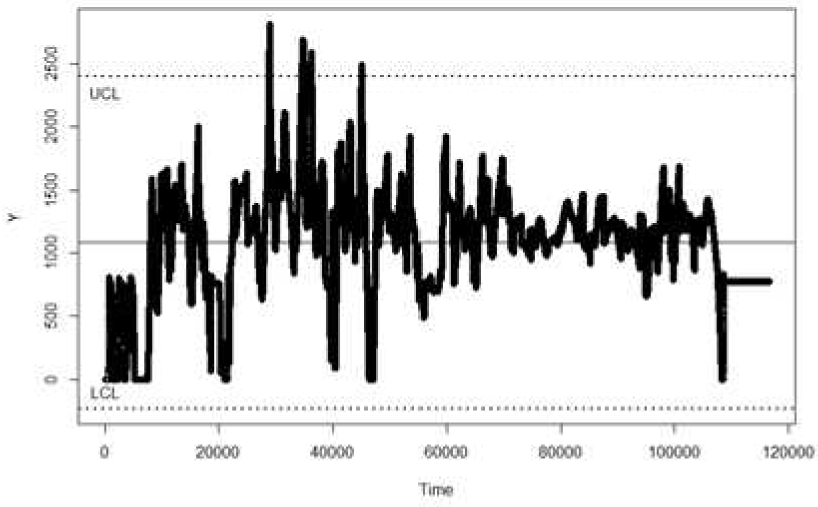
```
Joe.Markov.MLE(accel.pedal, plot=TRUE)
```



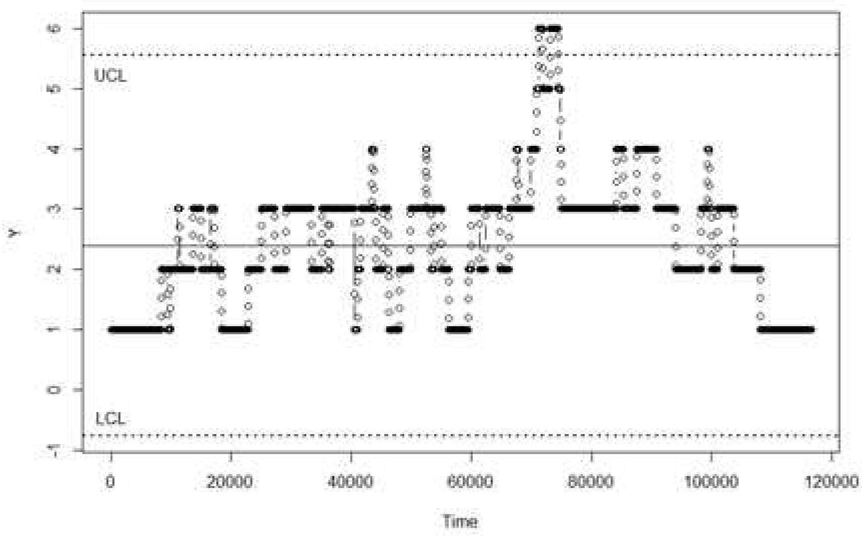

```
Joe.Markov.MLE(engine.rpm, plot=TRUE)
```



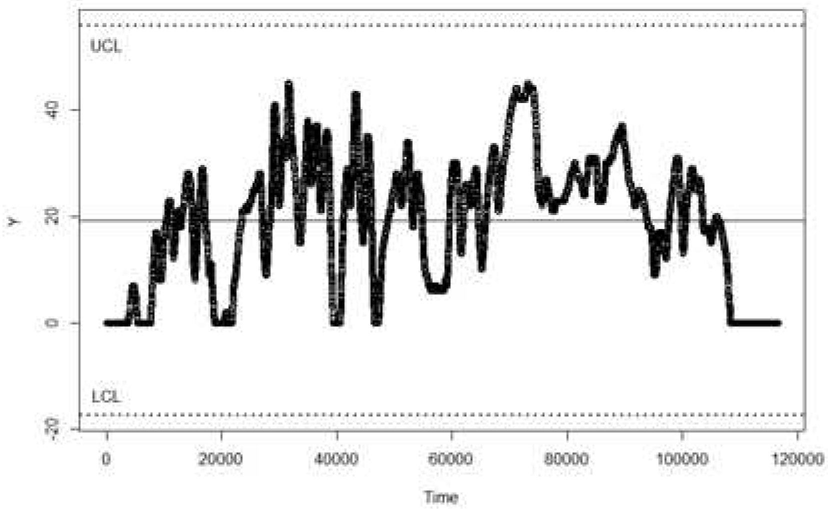

```
Joe.Markov.MLE(trans.rpm, plot=TRUE)
```



Joe.Markov.MLE(gear, plot=TRUE)



Joe.Markov.MLE(speed, plot=TRUE)



3. Principal Component Analysis

```
pr.out=prcomp(signal1, scale=TRUE) # standardized variables
pr.out$rotation
```

	PC1	PC2	PC3	PC4	PC5
accel.pedal	0.3451049	0.5363174	-0.73801261	-0.19818332	-0.09657615
engine.rpm	0.4779914	0.3898794	0.29006378	0.69336303	0.23372844
trans.rpm	0.5140277	0.1074649	0.47546717	-0.67883569	0.19323536
gear	0.3664350	-0.6312675	-0.38033424	0.07153022	0.56343233
speed	0.5039124	-0.3876987	0.02184564	0.11847621	-0.76239633

The k_th column is the k_th principal component score vector.

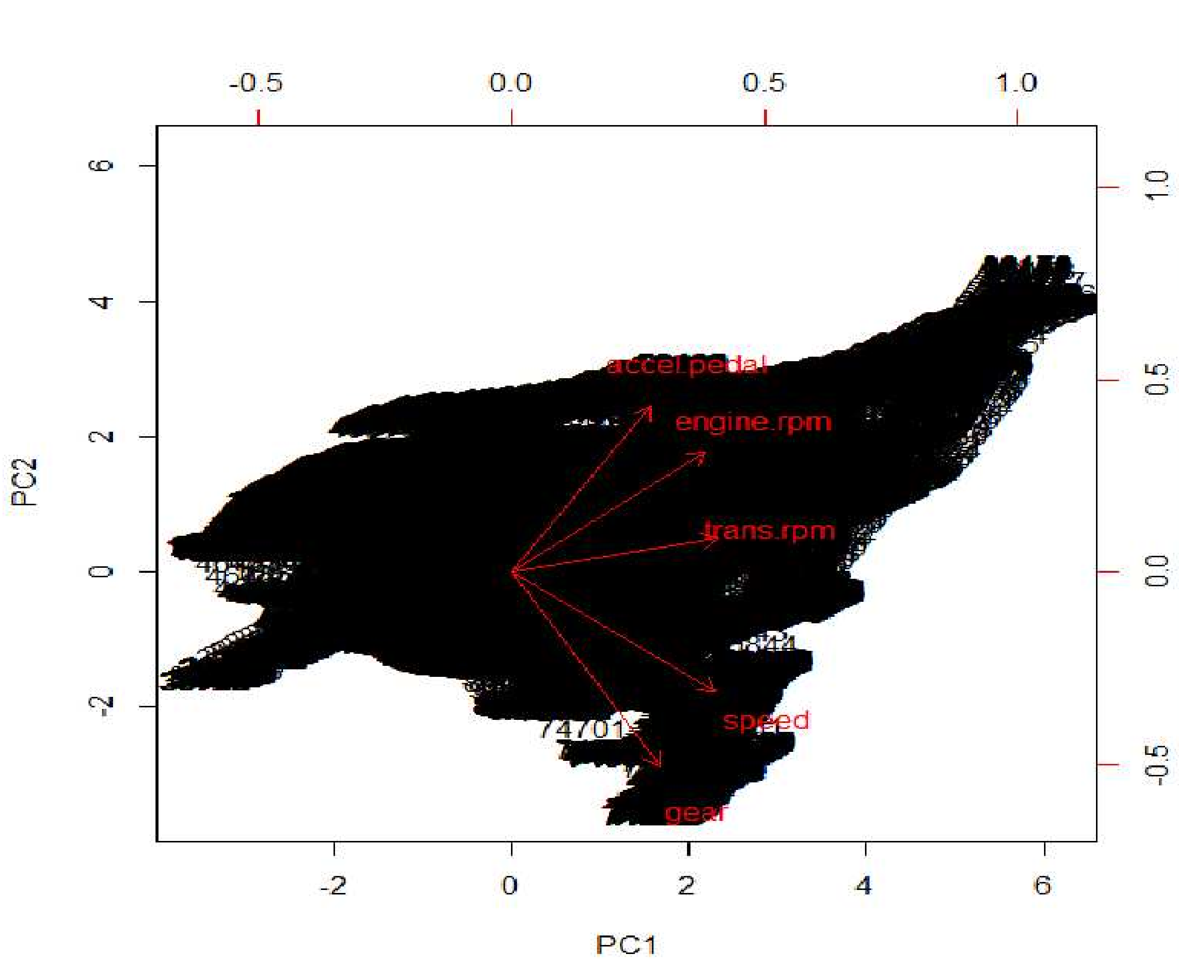
The first principal component scores are $z_{i1} = 0.345(x_{i1} - \overline{x}_1) + 0.478(x_{i2} - \overline{x}_2) + 0.514(x_{i3} - \overline{x}_3) + 0.366(x_{i4} - \overline{x}_4) + 0.504(x_{i5} - \overline{x}_5)$, i =1, 2, …, 116593

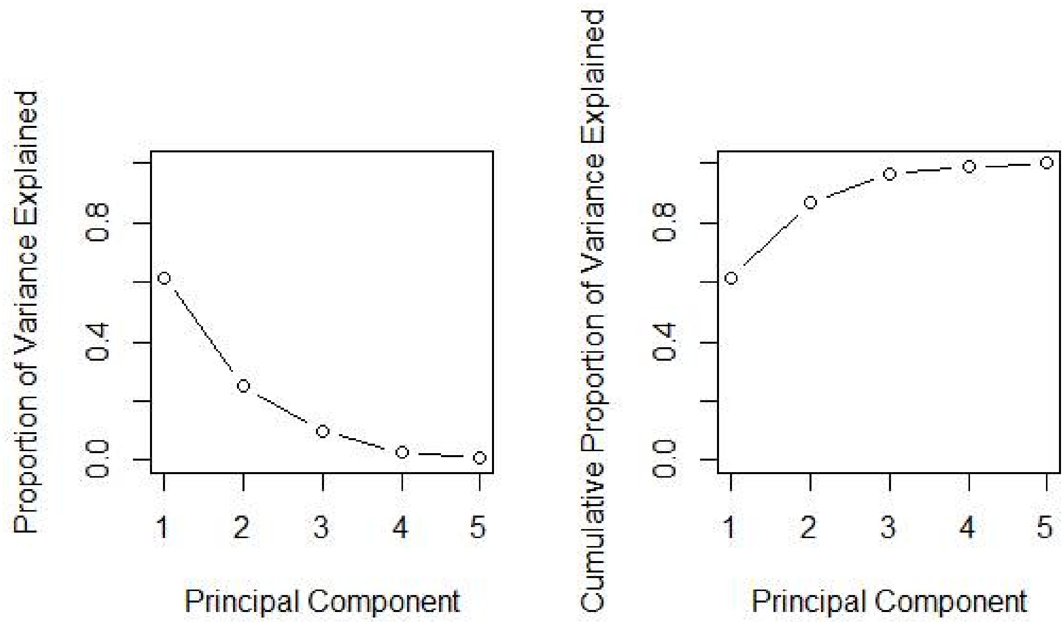
It looks like overall mean.

The second principal component scores are

$z_{i1} = 0.536(x_{i1} - \overline{x}_1) + 0.390(x_{i2} - \overline{x}_2) + 0.107(x_{i3} - \overline{x}_3) - 0.631(x_{i4} - \overline{x}_4) - 0.388(x_{i5} - \overline{x}_5)$, i =1, 2, …, 116593

```
biplot(pr.out, scale=0)
```

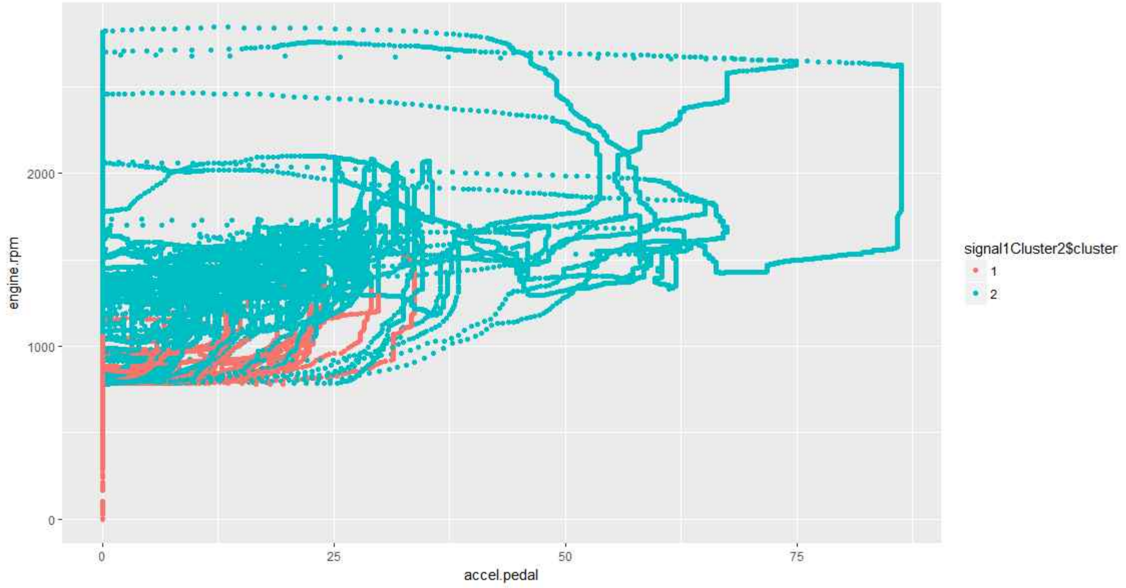




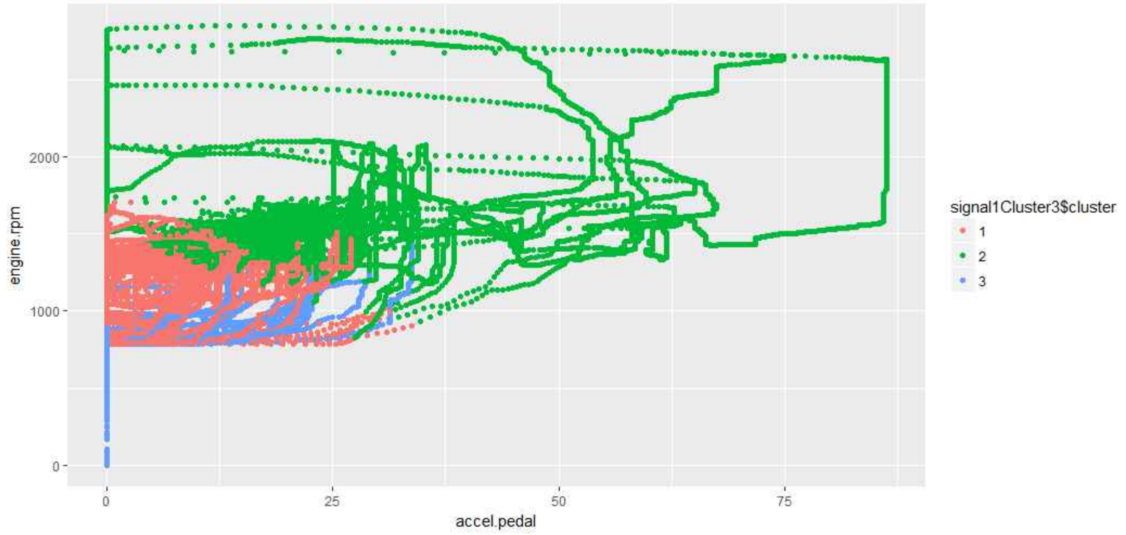
4. K-Means Clustering

4.1 accel.pedal against engine.rpm

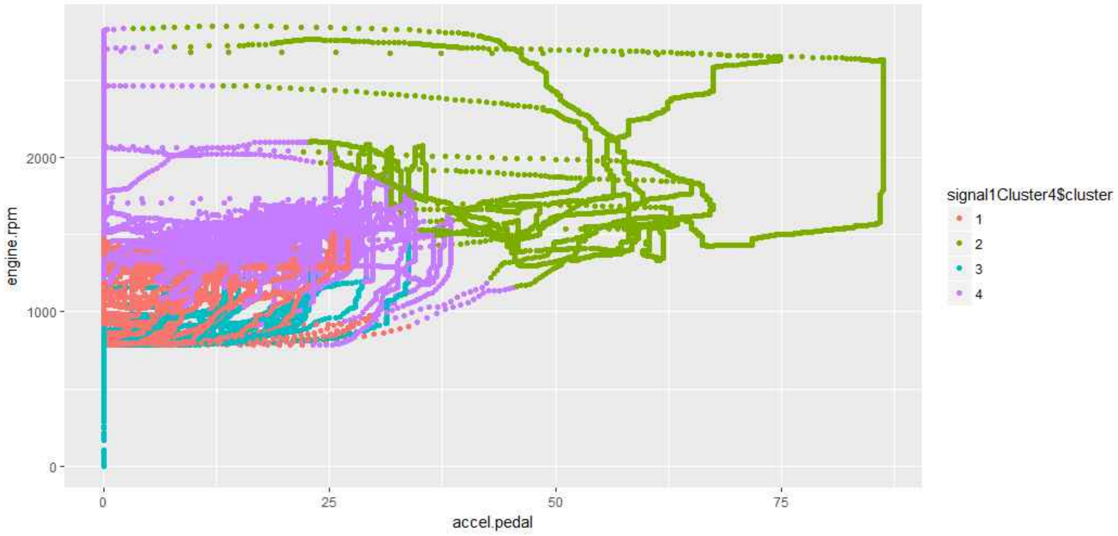
ggplot(signal1, aes(accel.pedal, engine.rpm, color = signal1Cluster2$cluster)) + geom_point()



ggplot(signal1, aes(accel.pedal, engine.rpm, color = signal1Cluster3$cluster)) + geom_point()

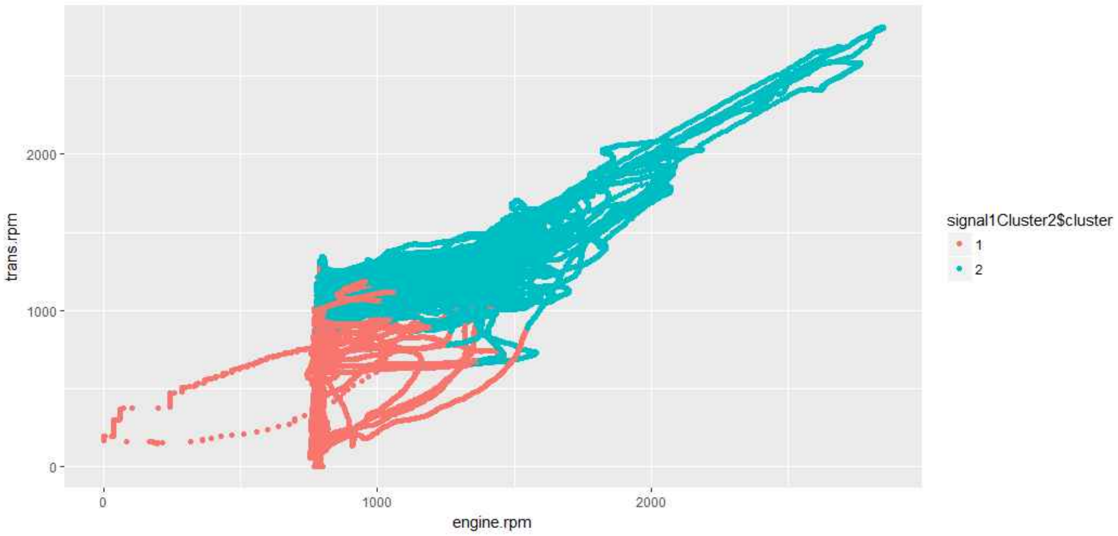


ggplot(signal1, aes(accel.pedal, engine.rpm, color = signal1Cluster4$cluster)) + geom_point()

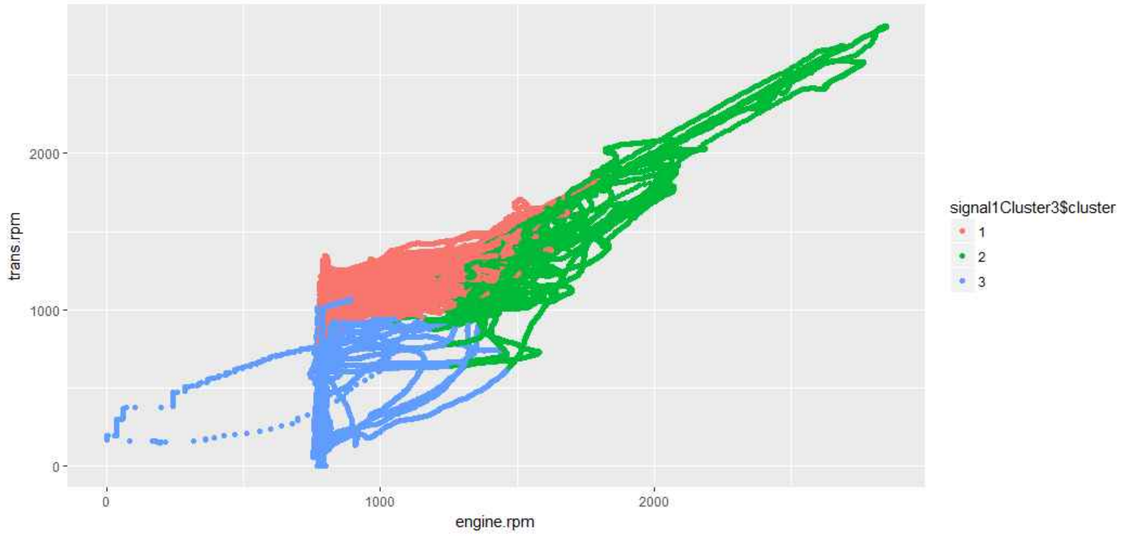


4.2 engine.rpm against trans.rpm

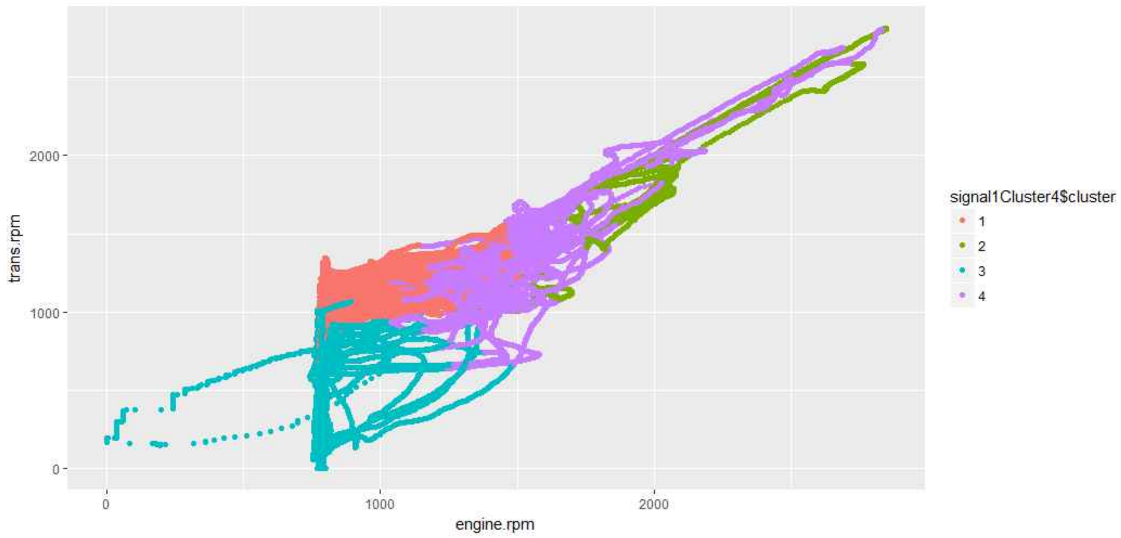
ggplot(signal1, aes(engine.rpm, trans.rpm, color = signal1Cluster2$cluster)) + geom_point()



ggplot(signal1, aes(engine.rpm, trans.rpm, color = signal1Cluster3$cluster)) + geom_point()

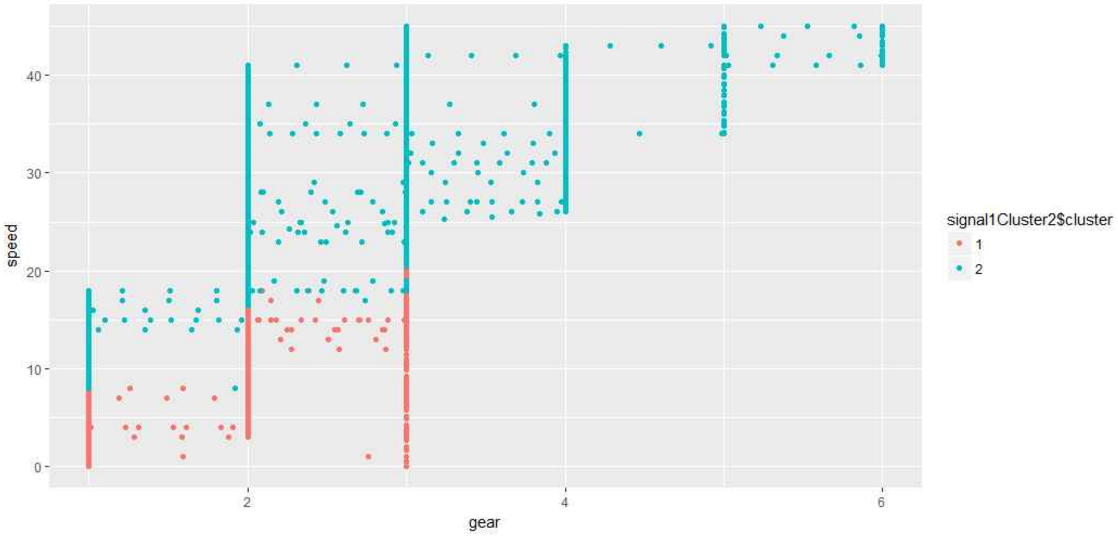


ggplot(signal1, aes(engine.rpm, trans.rpm, color = signal1Cluster4$cluster)) + geom_point()

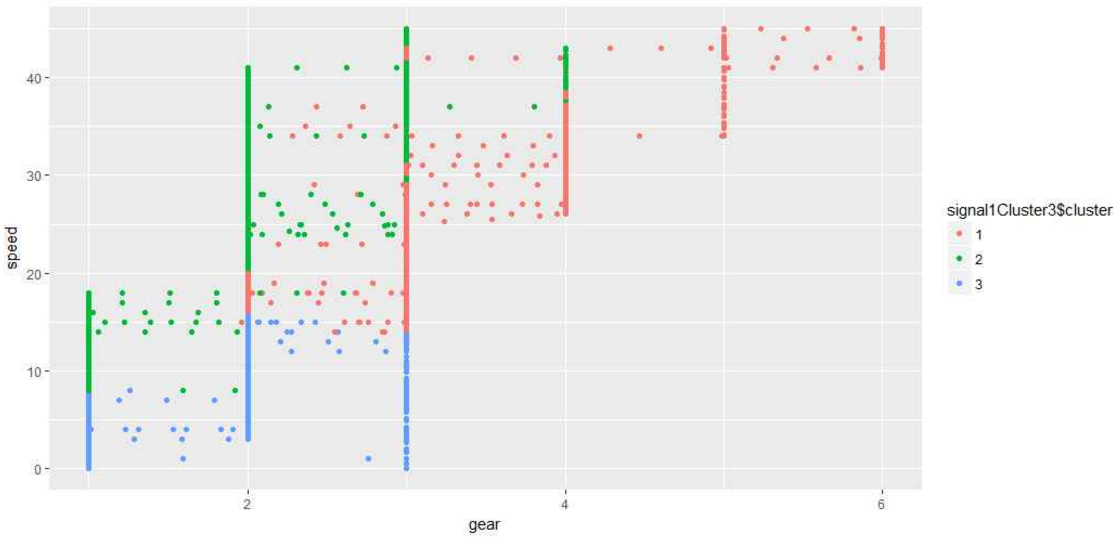


4.3 gear against speed

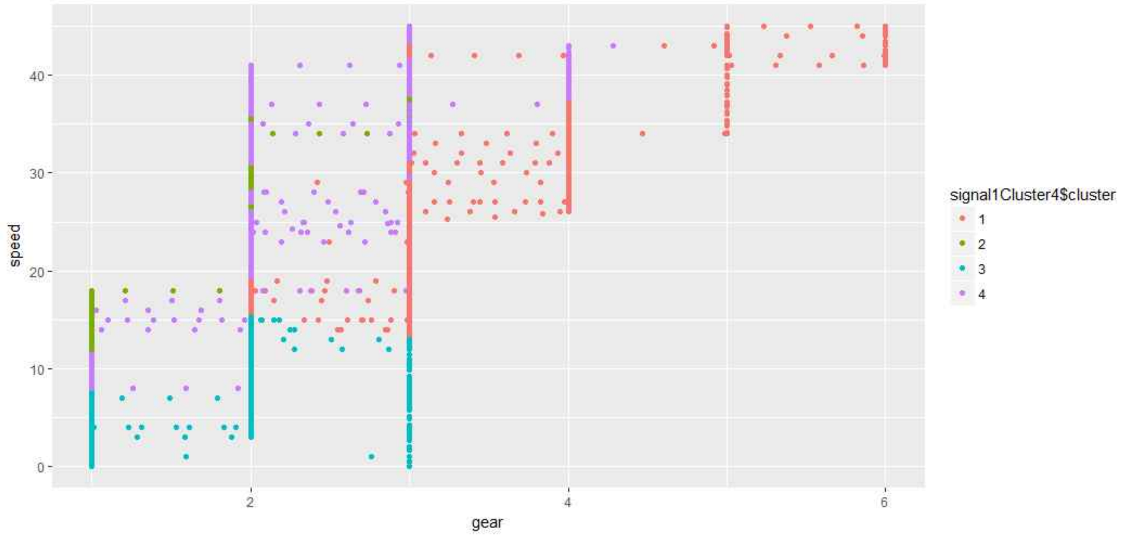
ggplot(signal1, aes(gear, speed, color = signal1Cluster2$cluster)) + geom_point()



ggplot(signal1, aes(gear, speed, color = signal1Cluster3$cluster)) + geom_point()

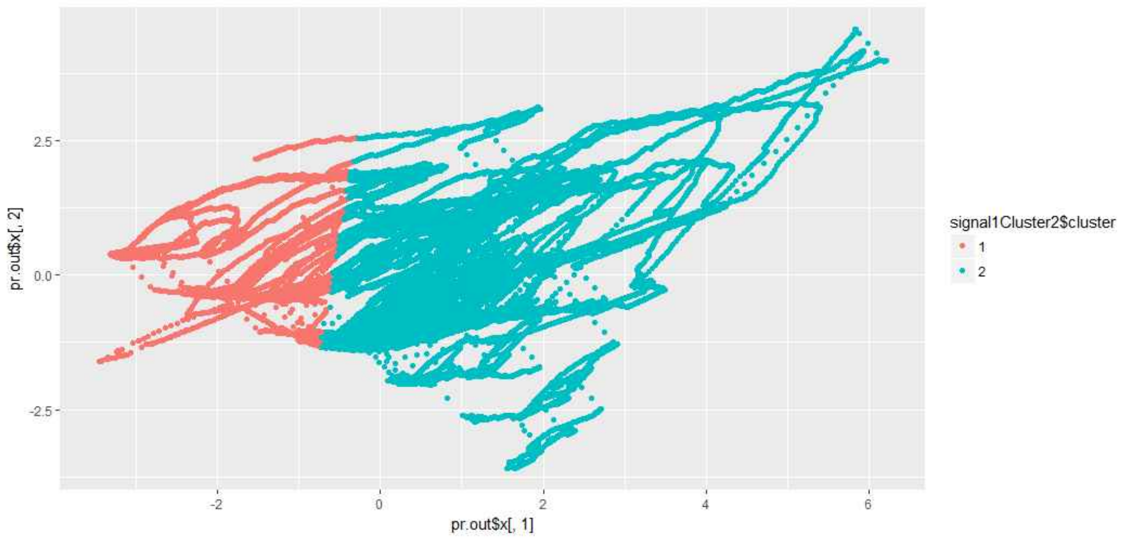


ggplot(signal1, aes(gear, speed, color = signal1Cluster4$cluster)) + geom_point()

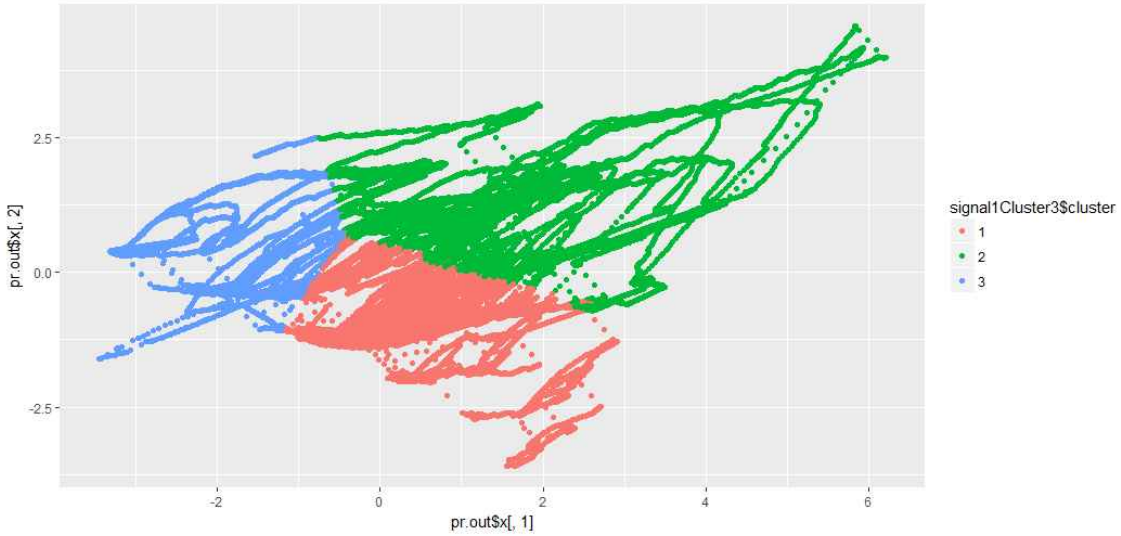


4.4 K-means clustering in terms of principal components 1 and 2

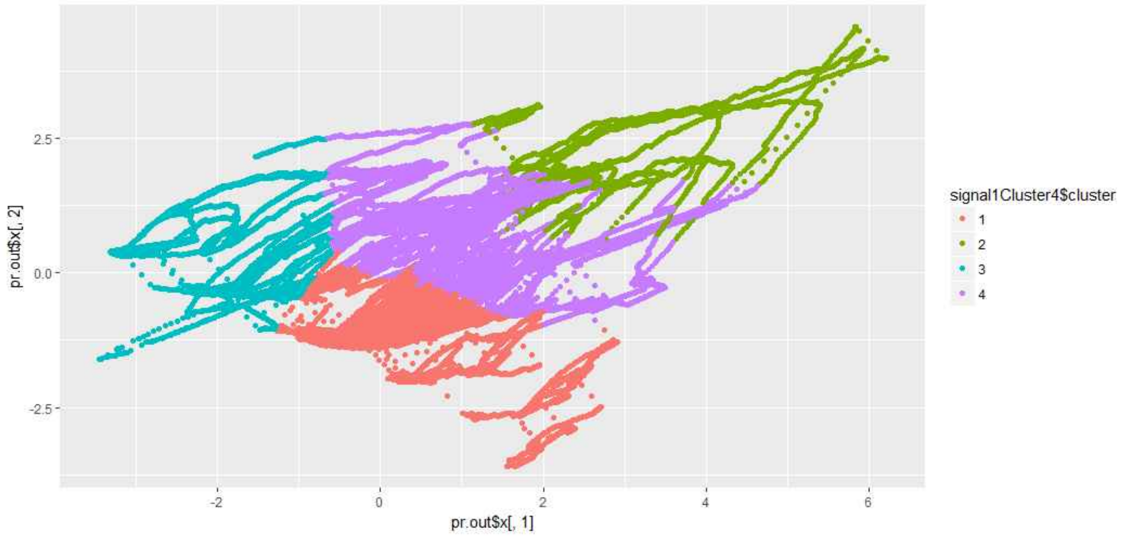
ggplot(signal1, aes(pr.out$x[,1], pr.out$x[,2], speed, color = signal1Cluster2$cluster)) + geom_point()



ggplot(signal1, aes(pr.out$x[,1], pr.out$x[,2], color = signal1Cluster3$cluster)) + geom_point()



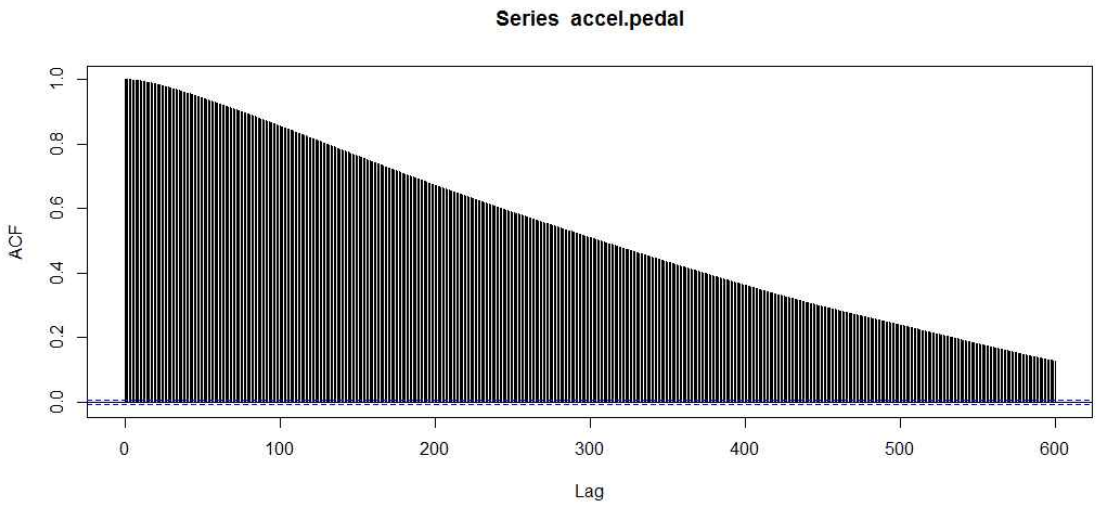

```
ggplot(signal1, aes(pr.out$x[,1], pr.out$x[,2], color = signal1Cluster4$cluster)) + geom_point()
```



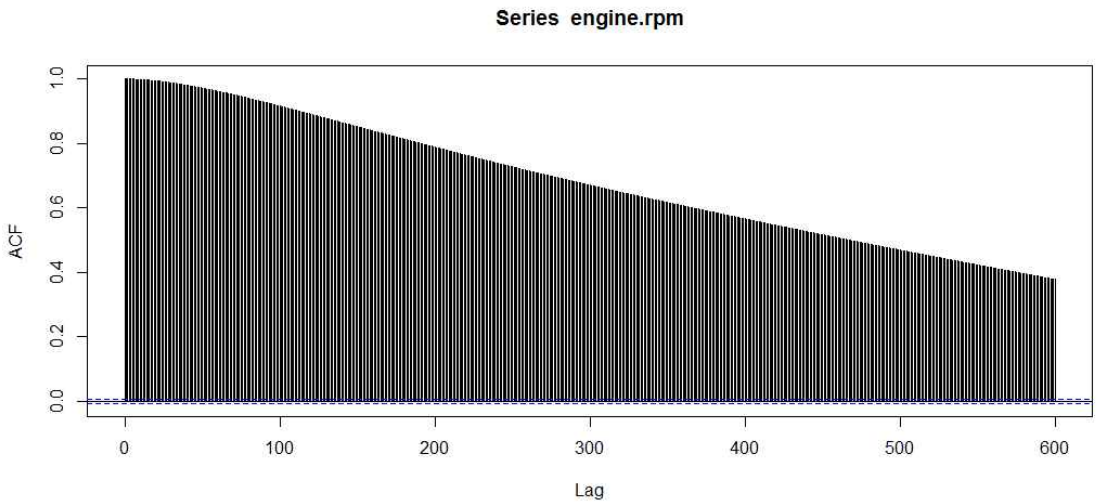
5. Time Series

5.1 Autocorrelation of each variable

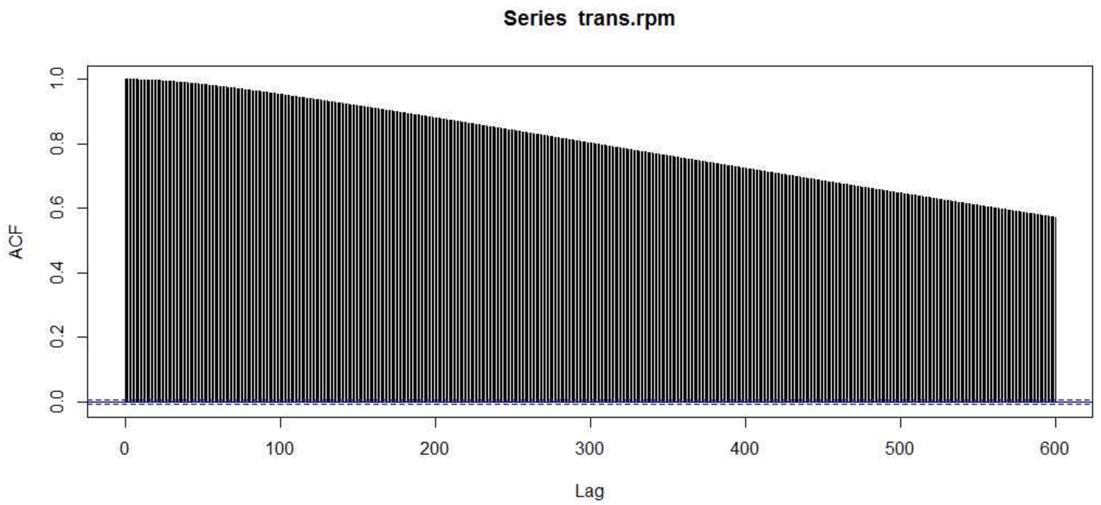
```
z1 = acf(accel.pedal, lag.max=600)
```



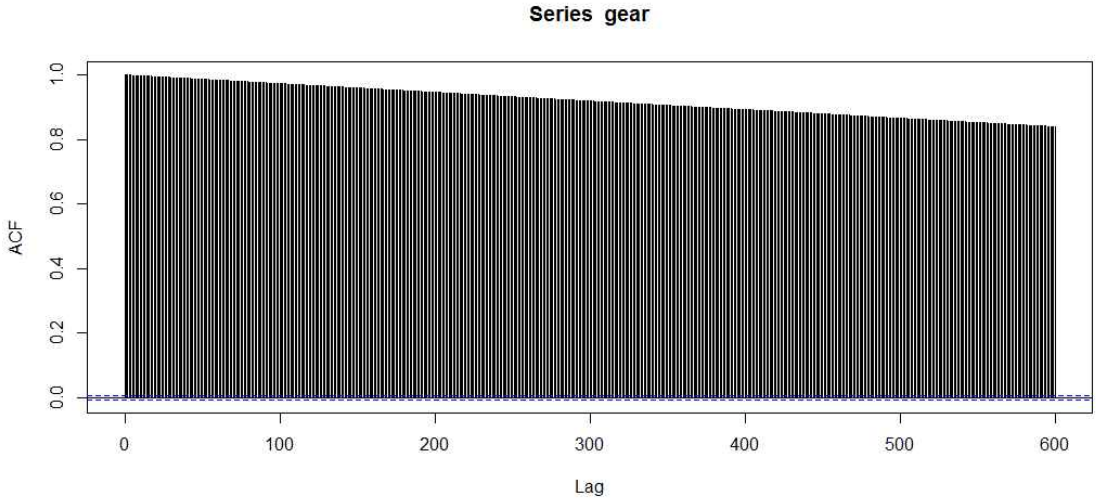
z2 = acf(engine.rpm, lag.max=600)



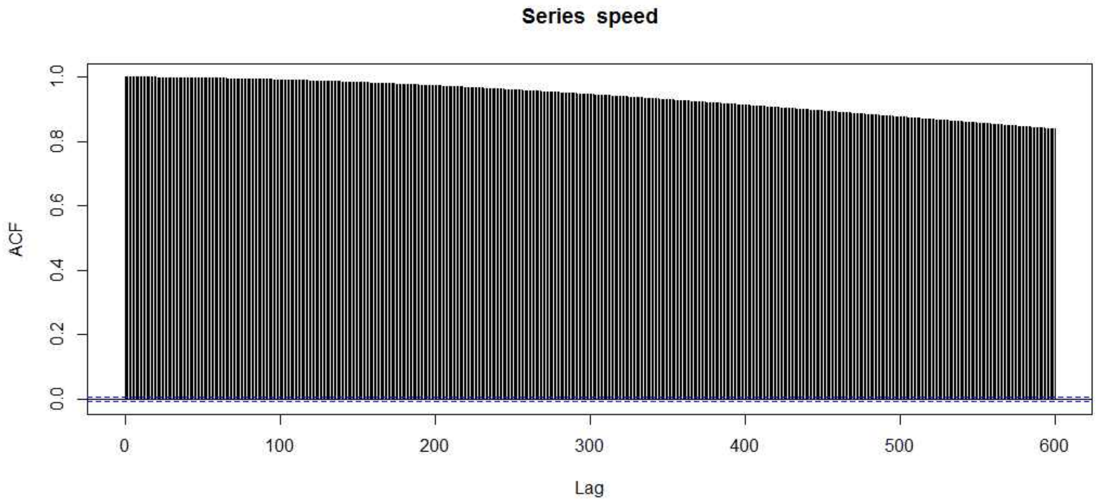
z3 = acf(trans.rpm, lag.max=600)



z4 = acf(gear, lag.max=600)

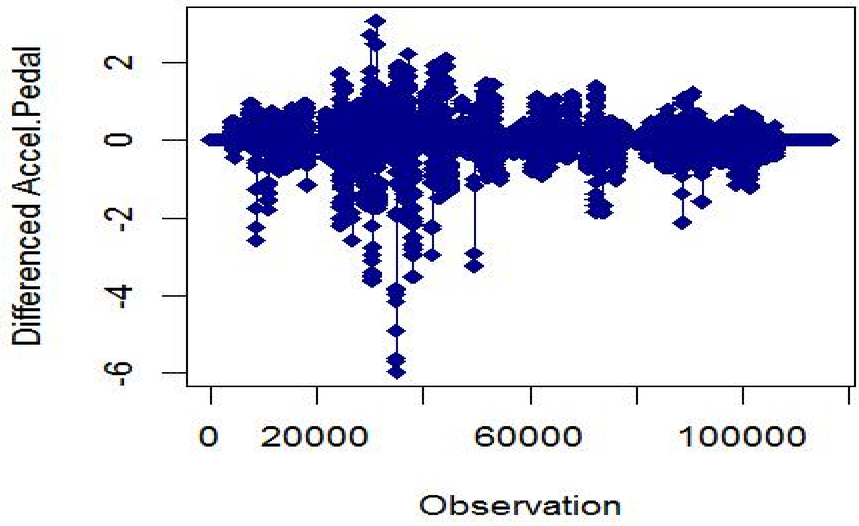



```
z5 = acf(speed, lag.max=600)
```

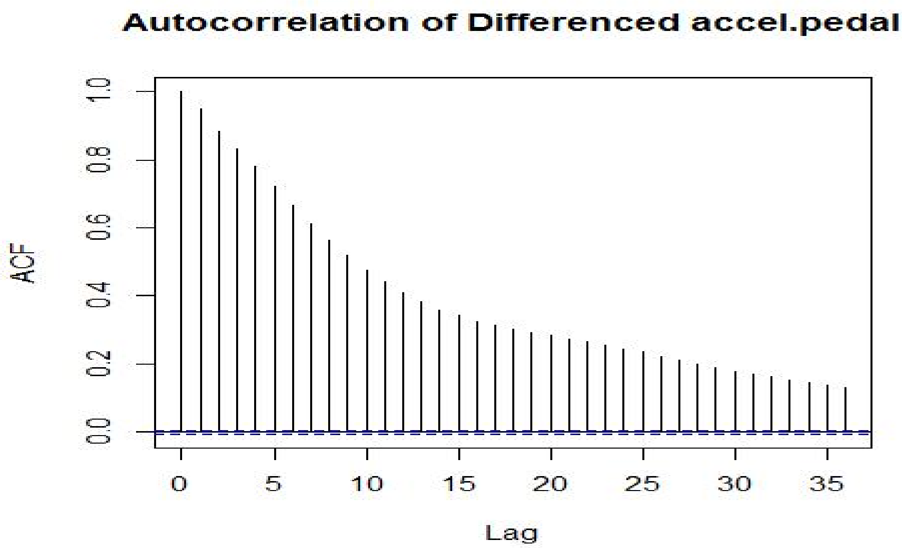


5.2 Modelling differenced series of accel.pedal

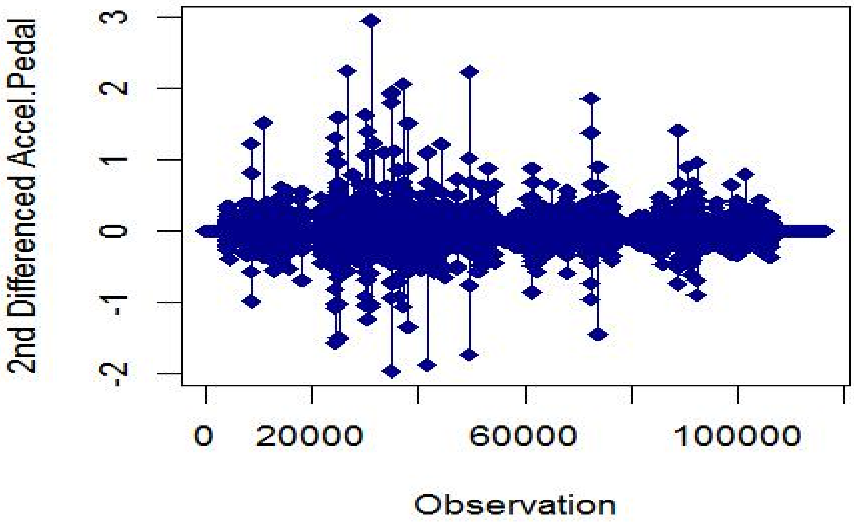
```
vec=ts(accel.pedal)
fd = diff(vec)
plot(fd, type="o", pch=18, col="darkblue",            xlab="Observation", ylab="Differenced
Accel.Pedal")
```



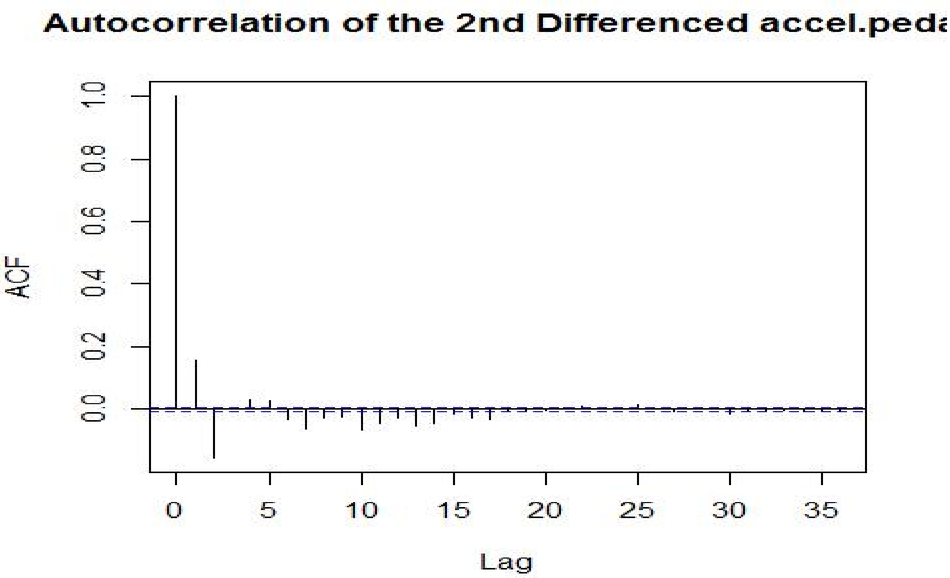
ac <- acf(fd, type = c("correlation"), lag.max=36, main="Autocorrelation of Differenced accel.pedal")



sd = diff(fd)
plot(sd, type="o", pch=18, col="darkblue", xlab="Observation", ylab="2nd Differenced Accel.Pedal")



ac <- acf(sd, type = c("correlation"), lag.max=36, main="Autocorrelation of the 2nd Differenced accel.pedal")



ma = arima(vec, order=c(0, 2, 2))
ma

Call:

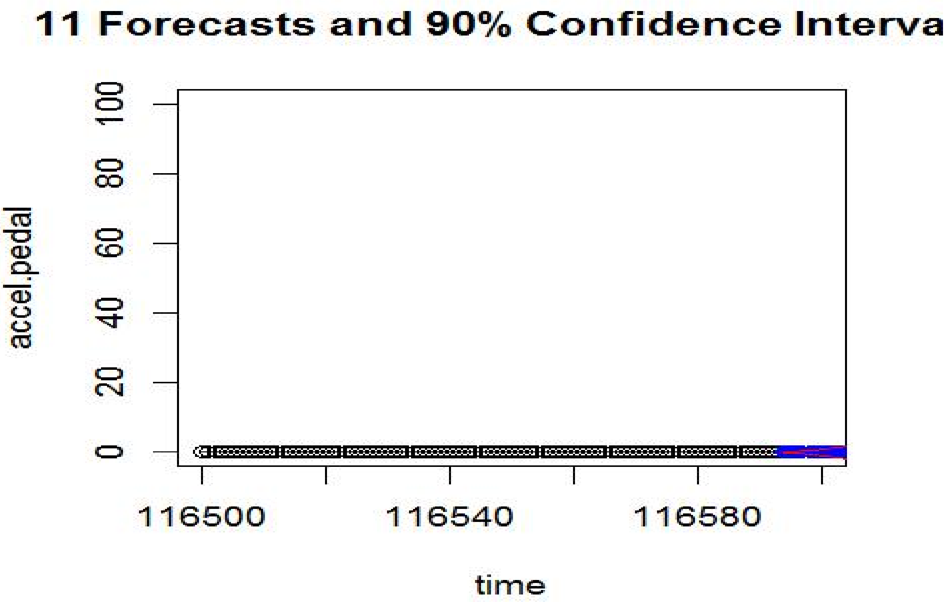
arima(x = vec, order = c(0, 2, 2))

Coefficients:

	ma1	ma2
	0.1992	-0.1588
s.e.	0.0029	0.0029

*sigma^2 estimated as **0.00201**: log likelihood = 196553.8, aic = -393101.6*

```
plot(c(116500:116593),accel.pedal[116500:116593],xlim=c(116500,116600),ylim=c(0,100),
type="o", ylab="accel.pedal", xlab="time", col="black", main="11 Forecasts and 90%
Confidence Intervals")
points(Forecast, pch=16, col="blue")
lines(c(116594:116605), L90, col="red")
lines(c(116594:116605), U90, col="red")
```



5.3 Multivariate Time Series

6. Dynamic Regression

Regress X5(Y) against X1, X2, X3, X4

3 min based dynamic regression

9095 points out of 95% CI

5107 points out of 99% CI

7. Literature Review

7.1 Detection algorithms for biosurveillance time series

Many Methods!

Method	Has Pitt/CMU tried it?	Tried but little used	Tried and used	Under development	Multivariate signal tracking?	Spatial ?
Time-weighted averaging	Yes	Yes				
Serfling	Yes		Yes			
ARIMA	Yes	Yes				
SARIMA + External Factors	Yes		Yes			
Univariate HMM	Yes		Yes			
Kalman Filter	Yes	Yes				
Recursive Least Squares	Yes		Yes			
Support Vector Machine	Yes	Yes				
Neural Nets	Yes	Yes				
Randomization	Yes		Yes	Yes		
Spatial Scan Statistics	Yes			(w/ Howard Burkom)	Yes	Yes
Bayesian Networks	Yes			Yes	Yes	
Contingency Tables	Yes		Yes			
Scalar Outlier (SQC)	Yes	Yes				
Multivariate Anomalies	Yes		Yes		Yes	
Change-point statistics	Yes			Yes		
FDR Tests	Yes		Yes		Yes	
WSARE (Recent patterns)	Yes		Yes	Yes	Yes	Yes
PANDA (Causal Model)	Yes			Yes	Yes	Yes
FLUMOD (space/Time HMM)				Yes	Yes	Yes

Details of these methods and bibliography available from "Summary of Biosurveillance-relevant statistical and data mining technologies" by Moore, Cooper, Tsui and Wagner. Downloadable (PDF format) from www.cs.cmu.edu/~awm/biosurv-methods.pdf

 Biosurveillance Detection Algorithms: Slide 2

7.2 Anomaly detection in streaming environmental sensor data

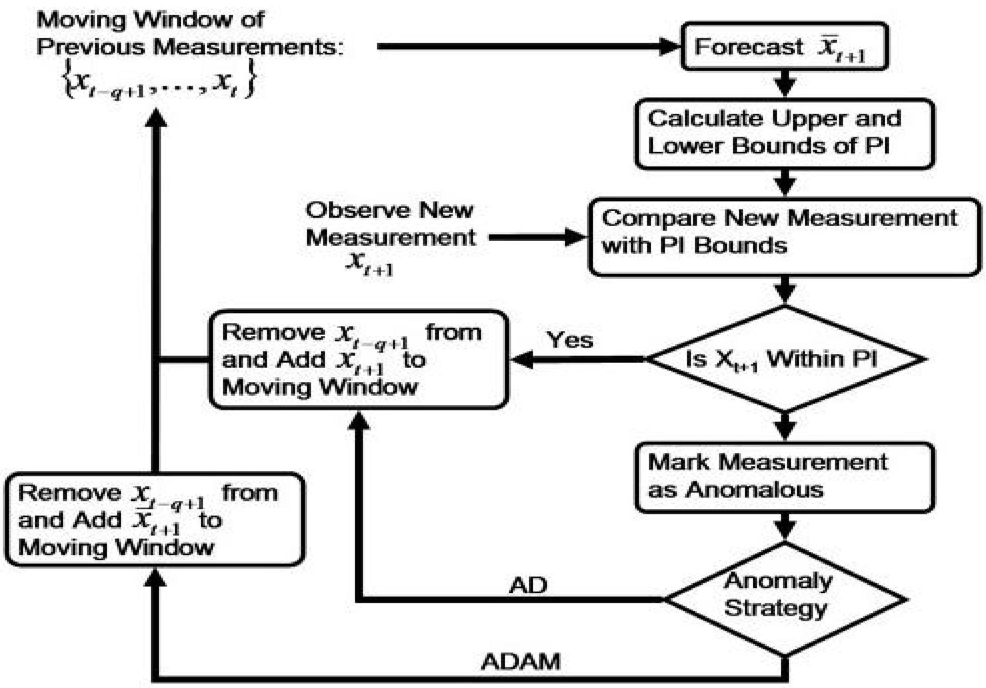


Fig. 1.
Schematic of proposed anomaly detection method.

Environmental Modelling &Software

Volume 25, Issue 9, September 2010, Pages 1014-1022

Anomaly detection in streaming environmental sensor data: A data-driven modeling approach

- David J. Hill [a], ,
- Barbara S. Minsker [b]

7.3 Intel Developer Zone

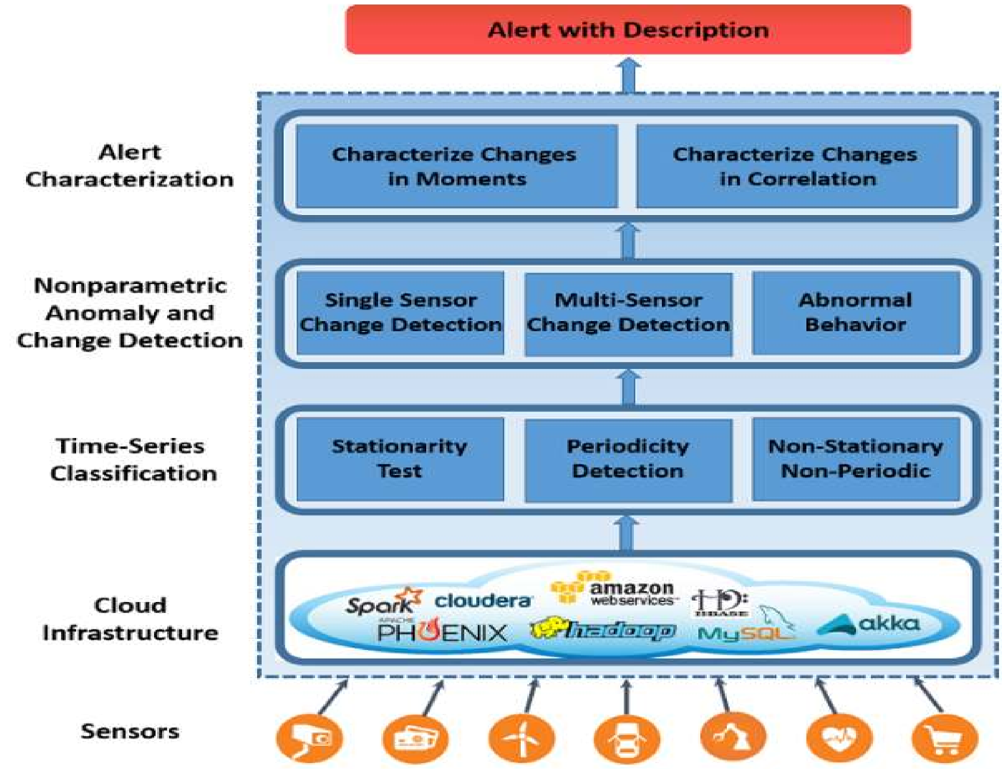


Figure 1: IoT Framework scheme: Our analytic engine consists of multiple layers including: sensor data ingestion and storage, time-series classification, anomaly and change detection, and alert characterization

intel Developer Zone

Development › Tools › Resources ›

Change and Anomaly Detection Framework for Internet of Things Data Streams

By Amitai A. (Intel), Gilad W. (Intel), Lev F. (Intel), Added June 17, 2016