# Test Data

|  |
| --- |
| The attached data represents issues raised in testing of a system that was done by 4 different drivers, in different locations and with different characteristics. The data was not processed and is presented as reported by the drivers. However the drives were evenly distributed on a 24hrs,each country and each road type. |
| Please answer the following questions: |
| 1. Which driver needs to change the way he characterize issues?   Ringo doesn't characterize 'minor' at all, meaning he probably radicalizes his severity assessment and can distribute the data in a more even manner. |
| 1. Which road type seems to be the most challenging While Urban roads have the highest percentage of critical severity (28%), Country and Highways both have a higher cumulative percentage of critical and major severity (around 50%). Although Urban's cumulative percentage of critical and major severity holds at around 45% (5% less than country and highways), I would classify Urban as the most challenging overall when taking into consideration it's high levels of critical severity, as well as its overall increasing percentage from critical until medium severity. |
| 3. In which country do we have the worst performance? Italy definitively has the highest percentage of critical and major severity rates. |
| 4. Which illumination has the best results?  Dusk has the highest percentage of minor severity rates. Although its critical severity rates are higher than Day and Night, the cumulative rates for critical and major together for Dusk are lower than the same cumulative rates for Day and Night.  Moreover, assuming the driving times were distributed evenly throughout the 24 hours, Dusk has the lowest overall rates of any kind of problematic driving. |

# Sensor

|  |
| --- |
| **Columns 1,2 and 3 represent 3 sensors outputs for a real world measure with a value of 2.47 (col 4) and a possible range of -2 to 6** |
| **1. Which is the best sensor in terms of signal quality and why?** I'd like to propose that sensor 3 is the sensor with the best signal quality. Although all three sensors average at ~2.47 (the real measure), series 3 has the smallest standard deviation and consistently stands at ~2.47. |
| **2. Which sensor is almost not useful at all and why?** I'd like to propose that series 1 is not useful. Although its average indeed is ~2.47, its standard deviation is the highest of the three series and it seems that the data is randomly distributed. |
| **3. What is the standard deviation of each data series?** a. Series1: 1.413749  b. Series2: 0.974244  c. Series3: 0.048712 |
| 4. What is the statistical distribution that best describes each sensor output? a. Series1: evenly random  b. Series2: normally distributed  c. Series3: normally distributed |
| 5. What is the RMS for each sensor?  a. Series1: 8.122096  b. Series2: 7.085643 c. Series3: 6.246673  Notes:   1. One can assume each measure of each sensor is independent of the other measures. 2. *If you are not familiar with any of the terms mentioned in the above questions try to search open sources on the internet.* |

# Parser

The file parser.log consists of a sensor with the following structure:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0x7E ~ | 0x40 @ | 0x51 Q | Data (17 bytes) | FCS(CRC) check sum | 0x7E ~ |

Inside the Data stream the speed is from the 9th bit to the 20th (including). The data is in little endian.

Whenever 7E should have appeared in the data or FCS calculation it was replaced by “7D 5E” {^ and 7D was replaced by “7D 5D”. {]

FCS is calculated as follows 0XFF -Sum(header and data bytes)%0X100

header and data bytes =all the message w/o starting and ending 0X7E and FCS.

The speed has a factor of 0.1 between the data and real value( meaning a value of 456 is actually 45.6)

Q:  
\*Note that my calculations were made in attached python script entitled parse\_log\_final.py and conclusions were drawn accordingly.

1. **The speed sensor has an undesired issue. Please define it**Every ~95 (+/-5) reads, the speed sensor gets an unexpected number (either above or below the current pattern [+/- 5]).   
   This was done by parsing each line according to the given structure and calculating the speed as mentioned above. I then did a quick manual onceover to detect any abnormalities.   
   I found that every ~95 reads, there speed jumps in any direction deviating from the orderly sequence.
2. **Find the wrong FCS**There was one wrong FCS in line 35 (in my version of parsing). This FCS contained an 'M' when it was supposed to contain '=' (4D instead of 3D).