OutlierAnalysis

Francis

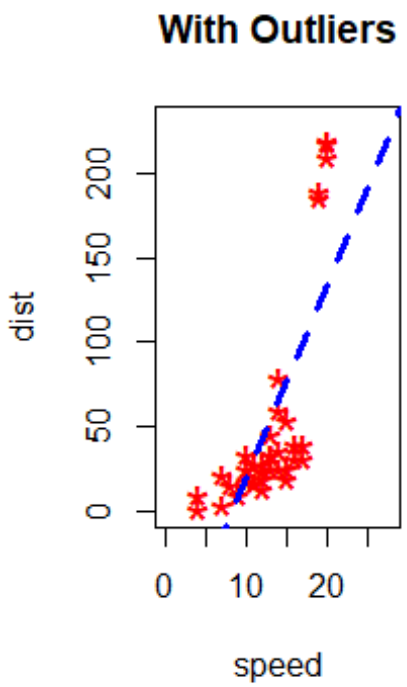
May 30, 2018

Outlier Treatment To investigate the importance of removing outliers we will add outliers to the cars dataset. Adding outliers

```
cars1 <- cars[1:30, ]  # original data
cars_outliers <- data.frame(speed=c(19,19,20,20,20), dist=c(190, 186, 210,
220, 218))  # introduce outliers.
cars2 <- rbind(cars1, cars_outliers)  # data with outliers.
```

Plot data with outliers. (use ggplot)

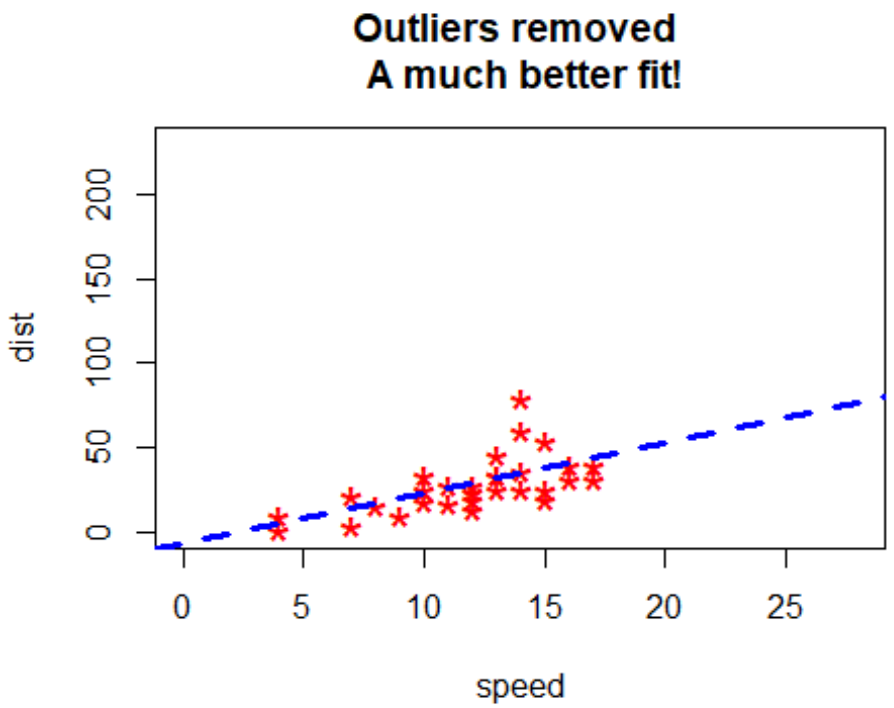
```
par(mfrow=c(1, 2)) #parse into 2 columns
plot(cars2$speed, cars2$dist, xlim=c(0, 28), ylim=c(0, 230), main="With
Outliers", xlab="speed", ylab="dist", pch="*", col="red", cex=2)
abline(lm(dist ~ speed, data=cars2), col="blue", lwd=3, lty=2)
```



Plot without outliers

```
plot(cars1$speed, cars1$dist, xlim=c(0, 28), ylim=c(0, 230), main="Outliers
removed \n A much better fit!", xlab="speed", ylab="dist", pch="*",
```

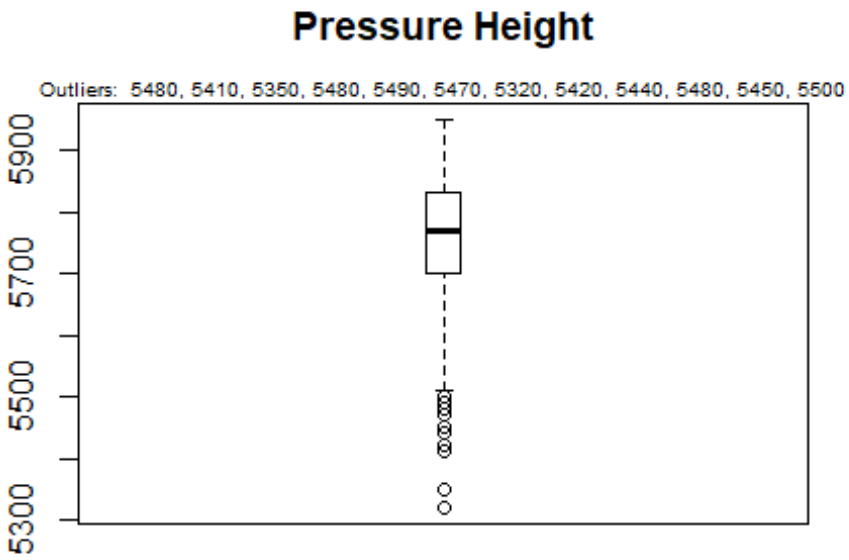
```
col="red", cex=2)
abline(lm(dist ~ speed, data=cars1), col="blue", lwd=3, lty=2)
```



Detecting outliers

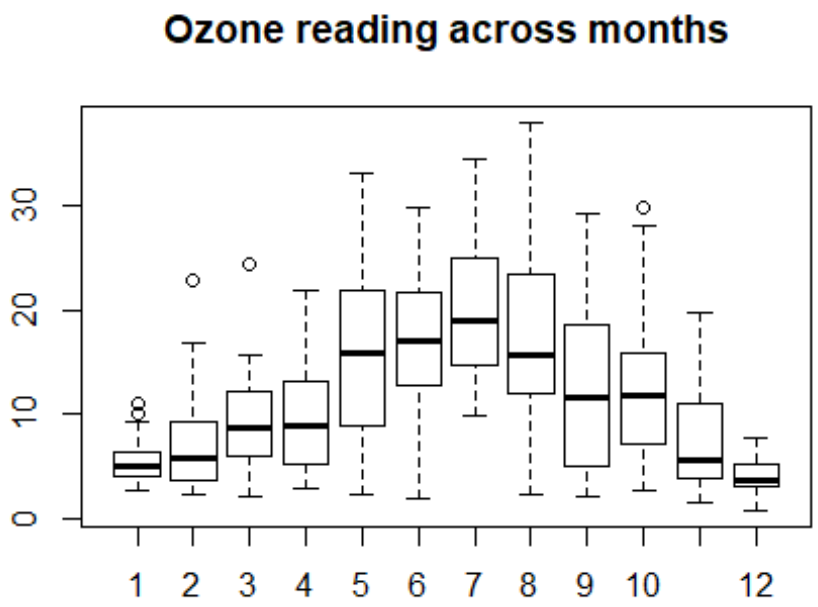
1. Univariate approach For a continuous variable, an outlier is considered to be an observation that lies outside 1.5*IQR or, the inter quartile range.

```
url <- "http://rstatistics.net/wp-content/uploads/2015/09/ozone.csv"
inputData <- read.csv(url)  # import data
outlier_values <- boxplot.stats(inputData$pressure_height)$out  # outlier
values.
boxplot(inputData$pressure_height, main="Pressure Height", boxwex=0.1)
mtext(paste("Outliers: ", paste(outlier_values, collapse=", ")), cex=0.6)
```

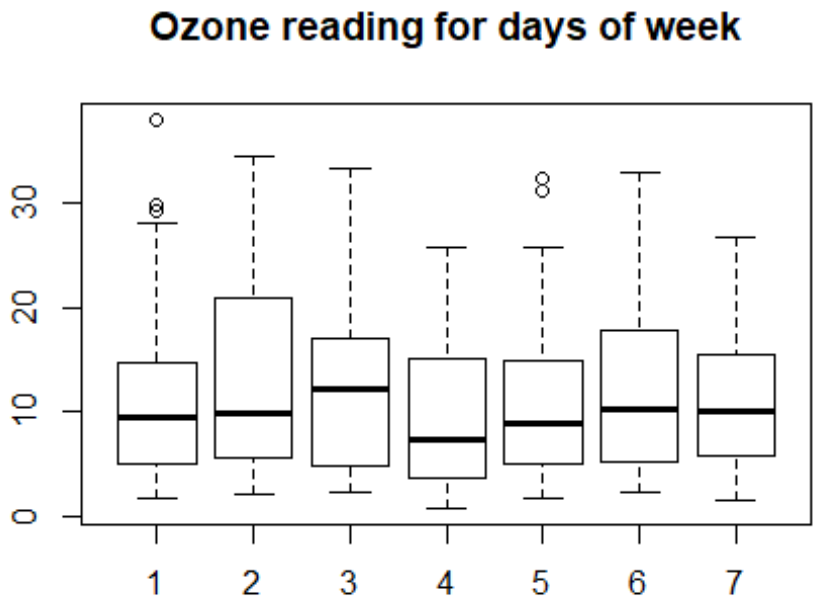


2. Bivariate approach visualize using box-plots of x and y for categorical X's

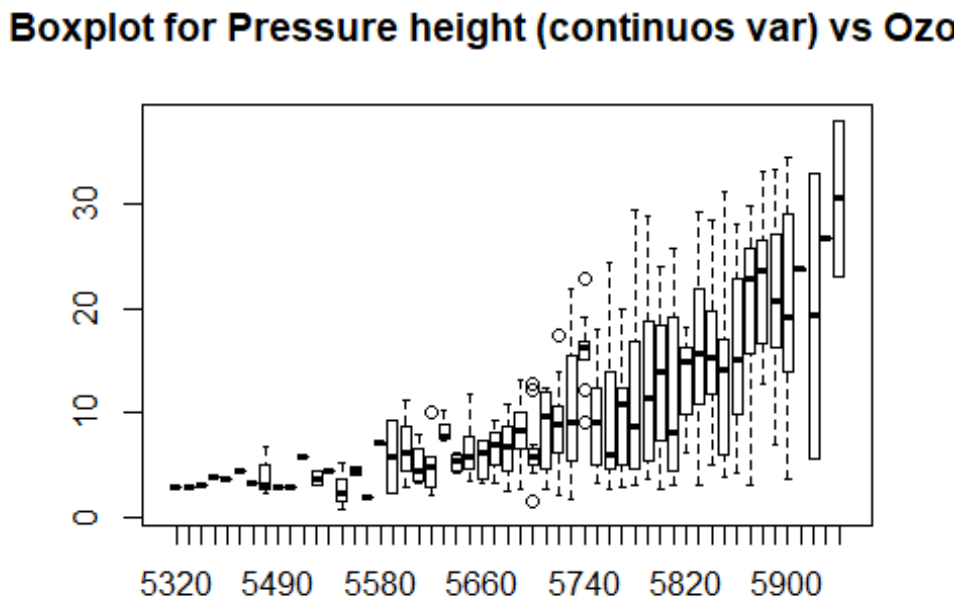
```
url <- "http://rstatistics.net/wp-content/uploads/2015/09/ozone.csv"
ozone <- read.csv(url)
# For categorical variable
boxplot(ozone_reading ~ Month, data=ozone, main="Ozone reading across
months")  # clear pattern is noticeable.
```



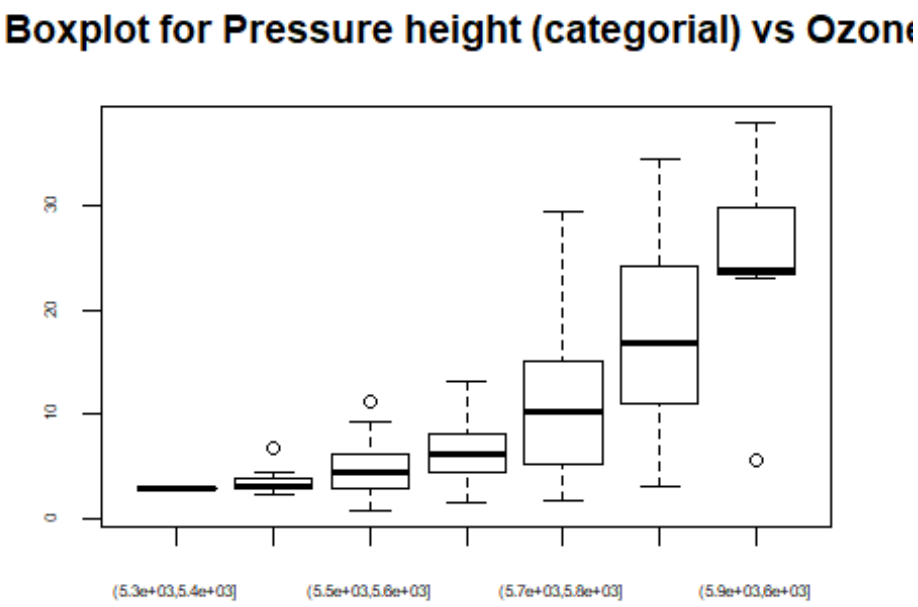

```
boxplot(ozone_reading ~ Day_of_week, data=ozone, main="Ozone reading for days
of week")
```



```r
# For continuous variable (convert to categorical if needed.)
boxplot(ozone_reading ~ pressure_height, data=ozone, main="Boxplot for
Pressure height (continuos var) vs Ozone")
```




```r
boxplot(ozone_reading ~ cut(pressure_height,
pretty(inputData$pressure_height)), data=ozone, main="Boxplot for Pressure
height (categorial) vs Ozone", cex.axis=0.5)
```

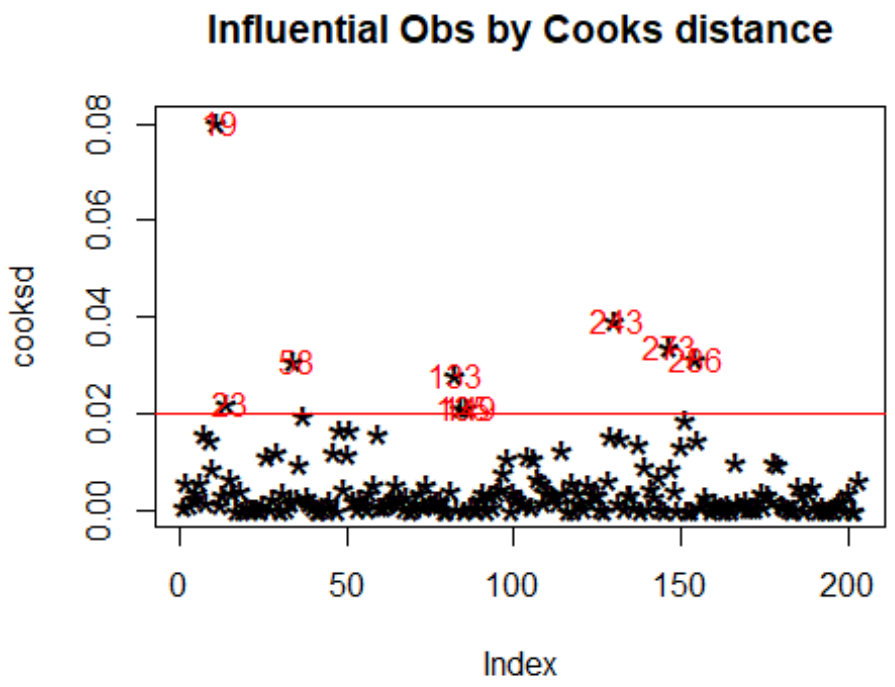


3. Multivariate approach Cook's distance: measures how much each observation impacts the fitted values

```
mod <- lm(ozone_reading ~ .,data = ozone)
cooksd <- cooks.distance(mod)
```

Observations with a cook's distance greater than 4 times the mean are influential.

```
plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks distance")  #
plot cook's distance
abline(h = 4*mean(cooksd, na.rm=T), col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd, labels=ifelse(cooksd>4*mean(cooksd,
na.rm=T),names(cooksd),""), col="red")  # add labels
```



identify infuential obs

```
influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=T))])
# influential row numbers
head(ozone[influential, ])  # influential observations.
```

```
##     Month Day_of_month Day_of_week ozone_reading pressure_height
## 19      1           19           1          4.07            5680
## 23      1           23           5          4.90            5700
## 58      2           27           5         22.89            5740
## 133     5           12           3         33.04            5880
## 135     5           14           5         31.15            5850
## 149     5           28           5          4.82            5750
##     Wind_speed Humidity Temperature_Sandburg Temperature_ElMonte
## 19           5       73                   52               56.48
## 23           5       59                   69               51.08
## 58           3       47                   53               58.82
## 133          3       80                   80               73.04
## 135          4       76                   78               71.24
## 149          3       76                   65               51.08
##     Inversion_base_height Pressure_gradient Inversion_temperature
## 19                    393               -68                 69.80
## 23                   3044                18                 52.88
## 58                    885                -4                 67.10
## 133                   436                 0                 86.36
## 135                  1181                50                 79.88
## 149                  3644                86                 59.36
```

```
##     Visibility
## 19          10
## 23         150
## 58          80
## 133         40
## 135         17
## 149         70
```

Outliers test

```
library("car")

## Loading required package: carData

library("outliers")
car::outlierTest(mod)

## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 243 3.045756          0.0026525      0.53845
```

Outliers Package outliers(): gets the most extreme observation from the mean.

```
set.seed(1234)
y=rnorm(100)
outlier(y)

## [1] 2.548991

#> [1] 2.548991
outlier(y,opposite=TRUE)

## [1] -2.345698

#> [1] -2.345698
dim(y) <- c(20,5)  # convert it to a matrix
outlier(y)

## [1] 2.415835 1.102298 1.647817 2.548991 2.121117

outlier(y,opposite=T)

## [1] -2.345698 -2.180040 -1.806031 -1.390701 -1.372302
```

scores(): computes the normalized scores and finds observations that lie outside a given percentile

```
set.seed(1000)
x = rnorm(10)
scores(x)
```

```
##  [1] -0.16730753 -1.25711119  0.53081842  1.38860927 -0.65591381
##  [6] -0.08086494 -0.21045016  1.50383307  0.44531788 -1.49693102
```

```
scores(x, type = "chisq")
```

```
##  [1] 0.027991810 1.580328533 0.281768196 1.928235712 0.430222922
##  [6] 0.006539139 0.044289270 2.261513911 0.198308018 2.240802490
```

```
scores(x, type = "t")
```

```
##  [1] -0.15798493 -1.30534811  0.50848339  1.47693426 -0.63373399
##  [6] -0.07626791 -0.19890433  1.63856418  0.42455313 -1.62854294
```

```
scores(x, type = "chisq", prob = 0.9)
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
scores(x, type="chisq", prob=0.95)  # beyond 95th %ile
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
scores(x, type="z", prob=0.95)  # beyond 95th %ile based on z-scores
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
scores(x, type="t", prob=0.95)  # beyond 95th %ile based on t-scores
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

Treating the outliers 1. Imputation same as above 2. Capping replacing values that lie outside 1.5*IQR with the value of the 5th percentile if the outlier is bellow the lower limit and the value of the 95th percentile if the outlier lies above the upper limit.

```
x <- ozone$pressure_height
qnt <- quantile(x, probs=c(.25, .75), na.rm = T)
caps <- quantile(x, probs=c(.05, .95), na.rm = T)
H <- 1.5 * IQR(x, na.rm = T)
x[x < (qnt[1] - H)] <- caps[1]
x[x > (qnt[2] + H)] <- caps[2]
```

3. Prediction See above