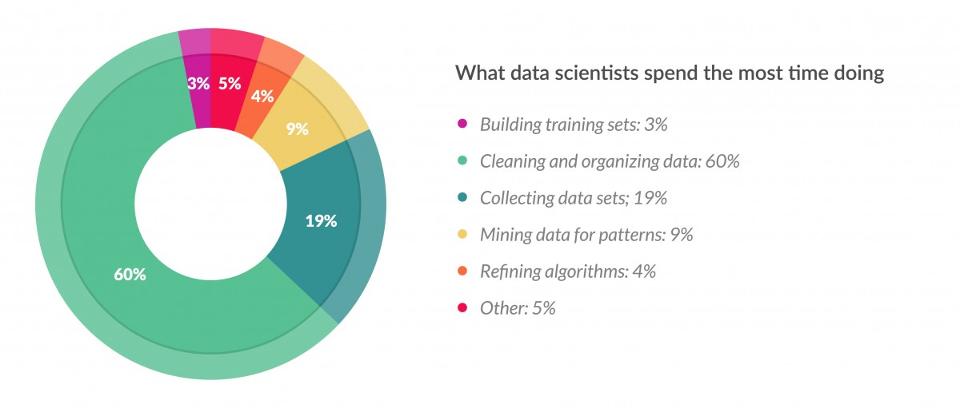
**Cleaning Raw Data**



Data scientists spends most of the time in cleaning their data than Mining it for Patterns

This is about 80% of their work.

### 

### **Cleaning Data :**

**1. Fixing the Formats :**

Often when data is saved or translated from one format to another some data may not be translated correctly like date or time stamp and sometime numbers may be stored as Characters so we have to correct them to their respective formats.

**Sample Code :**

For changing only date if it is in factor or character format we can use

**newDate <- as.Date(as.character(oldDate))**

Time stamp :

**newTime <- as.POSIXct(as.character(oldTime))** or **as.POSIXlt(as.character(oldTime))**

Some times format of the time stamp should be added for possixlt if time is not in standard format.

Ex : **Log2$V4 <- as.POSIXlt(Log2$V4,format =" %d/%b/%Y %H:%M:%S")**

**2. Filling in missing values :**

It is quite common for some values to be missing from datasets. This typically means that a piece of information was simply not collected. There are several options for handling missing data.

**A) Deleting the rows of missing values :**

If the no of rows having empty data in any coloumn is less than 10% of total rows we can omit these rows for cleaning our dataset.

**Sample Code :**

**newData <- oldData[complete.cases(oldData),]**

**B) Filling Data :**

If the no of rows having missing data is more than 10% then filling the data is required.

There are many ways to fill the missing data.

This depends on a range of factors, including the type of data we are trying to fill.

If the data is categorical (i.e. countries, device types, etc.), it may make sense to simply create a new category that will represent ‘unknown’. Another option may be to fill the values with the most common value for that column (the mode). However, because these are broad methods for filling the missing values, this may oversimplify your data and/or make your final model less accurate.

For numerical values (for example the age column) there are some other options. Given that in this case using the mode to fill values makes less sense, we could instead use the mean or median. We could even take an average based on some other criteria – for example filling the missing age values based on an average age for users that selected the same country\_destination.

**(i) Filling by linear regression :**

We can fill the values that are missed by taking a linear regression model if we know that our missing data in a particular column has nearly a linear relationship with any other coloumn variable.

If y[i] value is missing and we know that y and x has a near linear relationship then

**linearRegModel <- lm(y~x,Data=DataSet)**

This will give us the intercept and slopeof our model y = a+bx in a coefficients matrix

Coeff <- coefficients(linearRegModel)

**ypredicted <- Coeff[1] + Coeff[2] \* x[i]** #i is the index at the missed value

**(ii) Filling by Mean / Median / Mode :**

If few values are only missing then we can go for filling those values with Mean or Median or Mode of the remaining values

**(iii) Advanced Ways of Predicting :**

Prediction is most advanced method to impute your missing values and includes different approaches such as: kNN Imputation, rpart, and mice.

***.1 kNN Imputation :***

DMwR::knnImputation uses k-Nearest Neighbours approach to impute missing values. What kNN imputation does in simpler terms is as follows: For every observation to be imputed, it identifies ‘k’ closest observations based on the euclidean distance and computes the weighted average (weighted based on distance) of these ‘k’ obs.

The advantage is that you could impute all the missing values in all variables with one call to the function. It takes the whole data frame as the argument and you don’t even have to specify which variable you want to impute. But be cautious not to include the response variable while imputing, because, when imputing in test/production environment, if your data contains missing values, you won’t be able to use the unknown response variable at that time

**Sample Code :**

**library(DMwR)**

**Predictknn <- knnImputation(Aq,meth = "weighAvg")**

### ***.2 mice :***

mice short for Multivariate Imputation by Chained Equations is an R package that provides advanced features for missing value treatment. It uses a slightly uncommon way of implementing the imputation in 2-steps, using mice() to build the model and complete() to generate the completed data. The mice(df) function produces multiple complete copies of df, each with different imputations of the missing data. The complete() function returns one or several of these data sets, with the default being the first.

**Sample Code :**

**library(mice)**

**PredictMice <- complete(mice(Aq,method = "rf"))**

**C) Removing Outliers :**

Sometimes we have to remove certain outliers and false data suppose if we see sometimes while doing a survey on school kids or any they put age like 99 or any alphabet or any other random value as a symbol of NULL while collecting data for easiness of their job so when we look at it a school kid will not have the age of 99 or a special character so we have to remove these false data.

Also sometimes while analyzing the income of people in India many would fall in middle class and some will be very rich like Ambani brothers they are outliers and we have to remove them.

############ Cleaning Airquality Data available in R Example###############

**library(ggplot2)**

**data("airquality")**

**Aq <- airquality**

**Miss <- is.na(airquality)**

**nMiss <- sum(Miss)**

**count <- 0**

**for(i in 1:nrow(Aq)) {**

**for(j in 1:ncol(Aq)) {**

**if(is.na(Aq[i,j])==TRUE){**

**count <- count + 1**

**break**

**}**

**}**

**}**

**print(count)**

#count = 42 that is we have to delete 42 rows of data atleast to avoid missing values :/

#lets see where are the more no of missing values

**apply(Aq,2,function(colVar) sum(is.na(colVar)))**

#Ozone has 37 missing values

**prcntOfMissing <- (count)/nrow(Aq) \*100**

#gave around 28% are missing

**if(prcntOfMissing<10){**

**Aqcc = Aq[complete.cases(Aq),]**

**}**

#since percent is too high we cannot delete them

#So alternate approach of filling the data

**summary(Aq$Ozone)**

#Summary Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

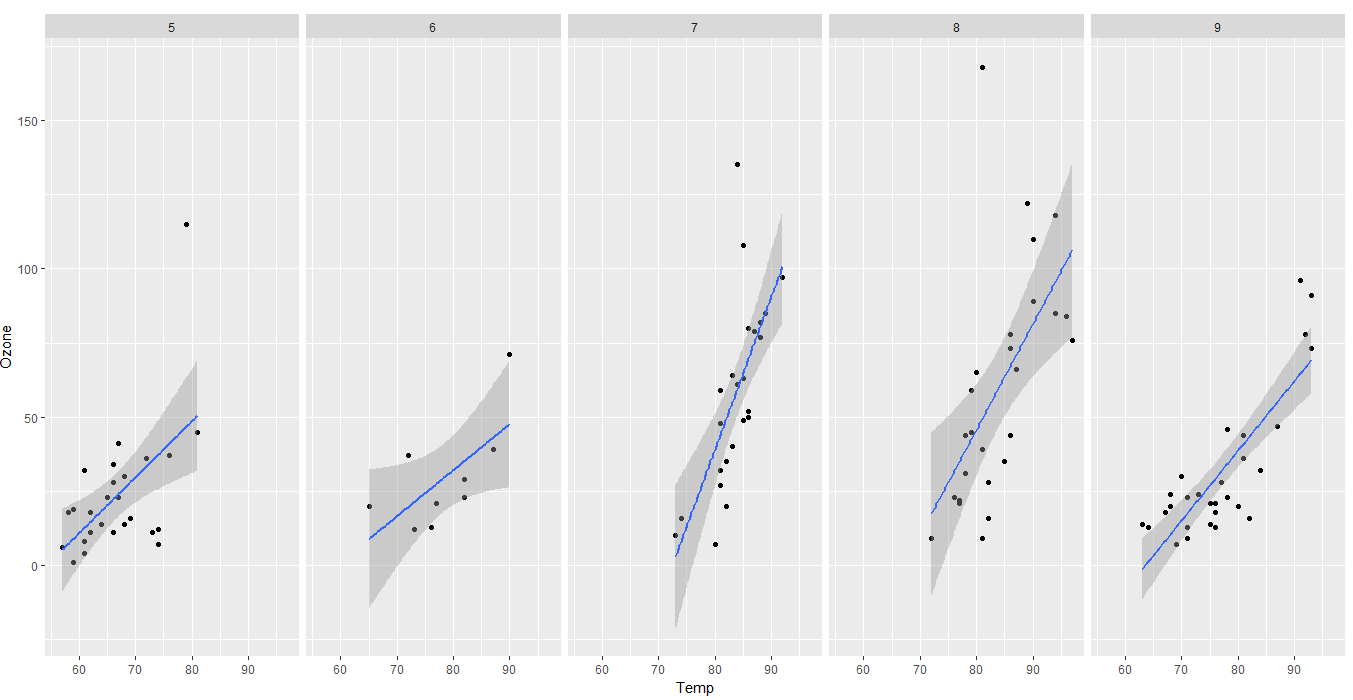
## 1.00 18.00 31.50 42.13 63.25 168.00 37

#Since research says that Ozone is dependent on Temperature or we can see the ozone dependencies on various other variables of our Data Set by visualizing.

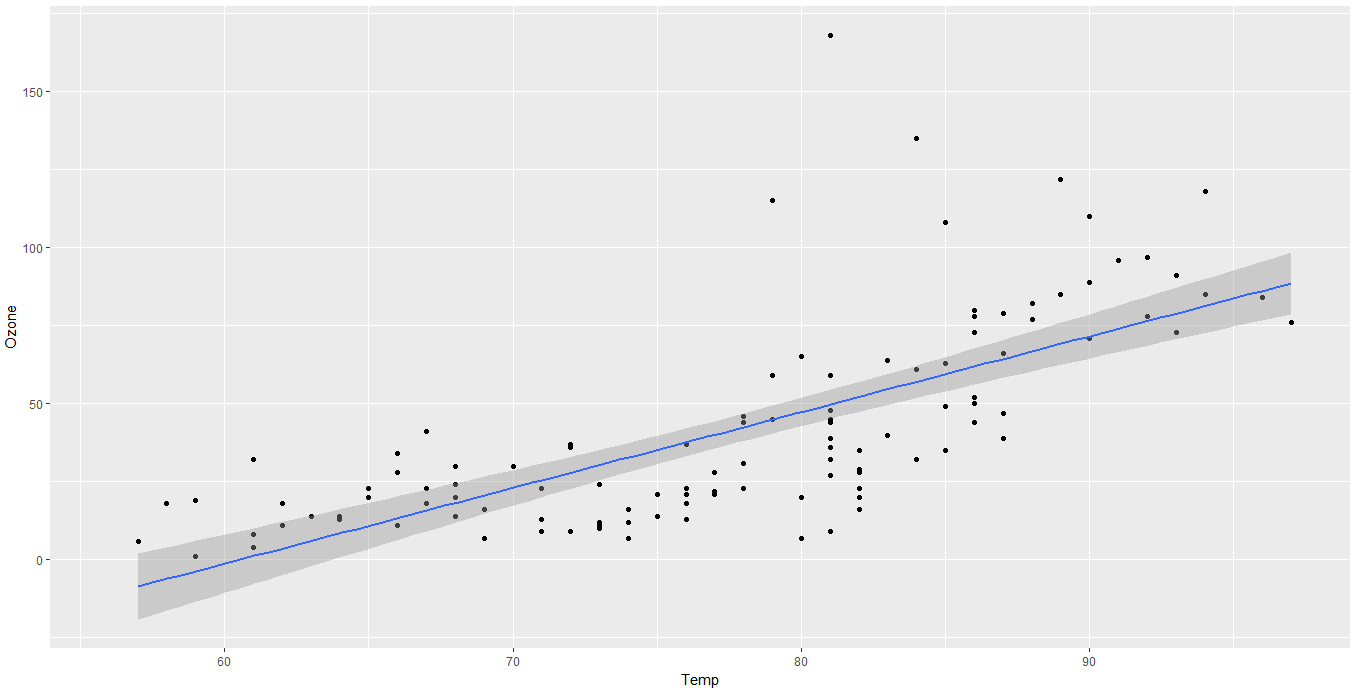
**ggplot(Aq,aes(Solar.R,Ozone))+geom\_point()+geom\_smooth(method = "lm")+facet\_grid(.~Month)** #no sign of dependency

**ggplot(Aq,aes(Wind,Ozone))+geom\_point()+geom\_smooth(method = "lm")+facet\_grid(.~Month)** #no sign of dependency

**ggplot(Aq,aes(Temp,Ozone))+geom\_point()+geom\_smooth(method = "lm")+facet\_grid(.~Month)**  # sign of dependency



**ggplot(Aq,aes(Temp,Ozone))+geom\_point()+geom\_smooth(method = "lm")** #overall dependency



**Month <- list()**

**for( j in seq(1:5)){**

**Month[[j]] <- subset(Aq,Month == j+4,select = c(Ozone,Temp,Month))**

**MothOzone.lm <- lm(Ozone~Temp,Month[[j]])**

**MonthCoeffs <- coefficients( MothOzone.lm)**

**predictMonth <- data.frame()**

**i = 1**

**for(i in seq(1:nrow(Month[[j]]))){**

**if(is.na(Month[[j]]$Ozone[i])==T){**

**predictMonth[i,1] <- MonthCoeffs[1]+MonthCoeffs[2]\*Month[[j]]$Temp[i]**

**}else {**

**predictMonth[i,1] = Month[[j]]$Ozone[i]**

**}**

**}**

**Month[[j]] <- cbind(Month[[j]],predictMonth)**

**}**

**Aqcc <- data.frame()**

**for(k in seq(1:5)){**

**Aqcc <- rbind(Aqcc,Month[[k]])**

**}**

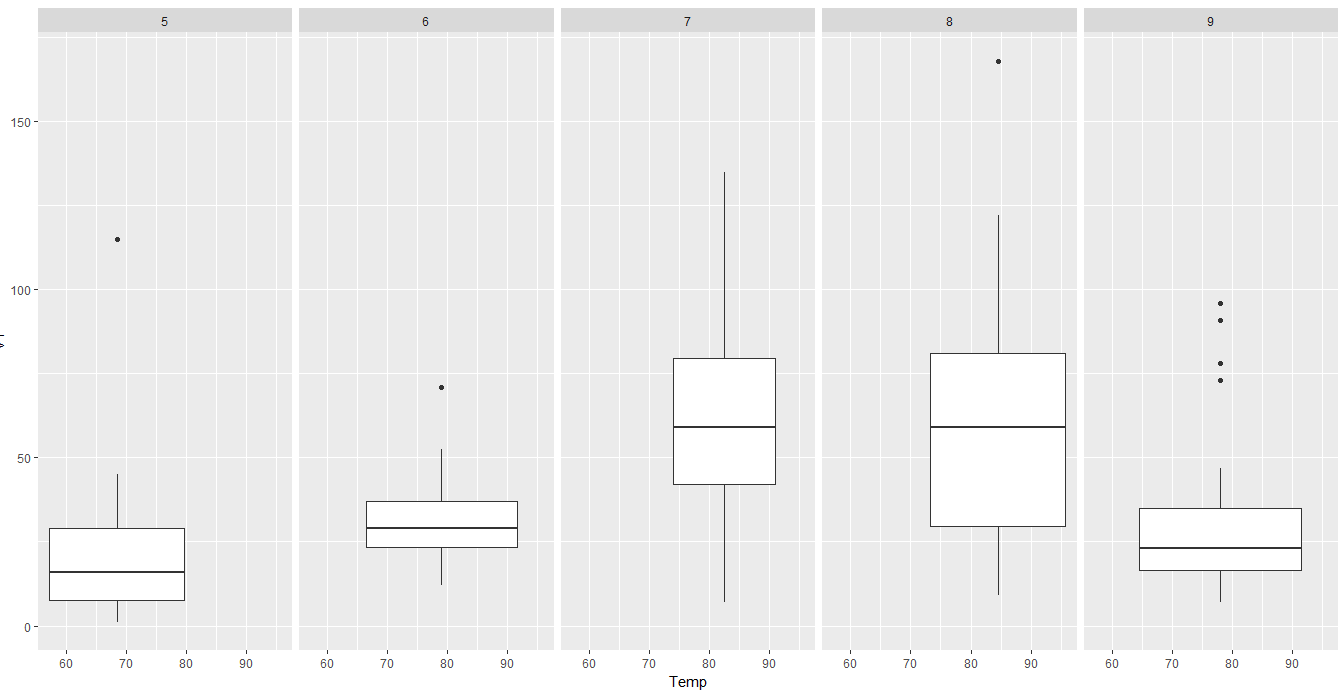
**summary(Aqcc$V1)**

# Min. 1st Qu. Median Mean 3rd Qu. Max.

# 1.00 20.00 31.00 40.69 52.39 168.00

**ggplot(Aq,aes(Temp,Ozone))+geom\_boxplot()+facet\_grid(.~Month)#viewing Outliers**

**ggplot(Aqcc,aes(Temp,V1))+geom\_boxplot(aes(group=1))+facet\_grid(.~Month)**



####################### Knn Imputaion #########################

**library(DMwR)**

**Predictknn <- knnImputation(Aq,meth = "weighAvg")**

################# MIce #########################################

**library(mice)**

**PredictMice <- complete(mice(Aq,method = "rf"))**

**regr.eval(Aqcc$V1,PredictMice)**