# Data Preparation and Preprocessing

> #import datasets

> crime12to17 <- read.csv('Desktop/7275/Chicago\_Crimes\_2012\_to\_2017.csv')

> head(crime12to17,3)



> #delete the data of 2017

> crime12to16 <- crime12to17[crime12to17$Year!='2017',]

> #delete the columns that are useless

> crime <- crime12to16[,-23]

> crime <- crime[,-12:-20]

> crime <- crime[,-6]

> crime <- crime[,-3]

> crime <- crime[,-1]

> head(crime,3)



> #creat col date

> library(lubridate)

> crime$Date <- as.Date(crime$Date, "%m/%d/%Y %I:%M:%S %p")

> crime$Day <- factor(day(as.POSIXlt(crime$Date, format="%m/%d/%Y %I:%M:%S %p")))

> crime$Month <- factor(month(as.POSIXlt(crime$Date, format="%m/%d/%Y %I:%M:%S %p"), label = TRUE))

> crime$Year <- factor(year(as.POSIXlt(crime$Date, format="%m/%d/%Y %I:%M:%S %p")))

> crime$Weekday <- factor(wday(as.POSIXlt(crime$Date, format="%m/%d/%Y %I:%M:%S %p"), label = TRUE))

> #recategory types

> length(unique(crime$Primary.Type))

[1] 33

> crime$types <- as.character(crime$Primary.Type)

> crime$types <- ifelse(crime$types %in% c("THEFT","MOTOR VEHICLE THEFT"), "THEFT", crime$types)

> crime$types <- ifelse(crime$types %in% c("NARCOTICS","OTHER NARCOTIC VIOLATION"), "NARCOTICS", crime$types)

> crime$types <- ifelse(crime$types %in% c("PROSTITUTION","CRIM SEXUAL ASSAULT", "SEX OFFENSE","OBSCENITY"), "SEXUAL OFFENCE", crime$types)

> crime$types <- ifelse(crime$types %in% c("OTHER OFFENSE","OFFENSE INVOLVING CHILDREN"), "OFFENSE", crime$types)

> crime$types <- ifelse(crime$types %in% c("NON - CRIMINAL","NON-CRIMINAL","NON-CRIMINAL (SUBJECT SPECIFIED)"), "NON-CRIMINAL", crime$types)

> crime$types <- ifelse(crime$types %in% c("STALKING","PUBLIC INDECENCY"), "STALKING", crime$types)

> crime$types <- ifelse(crime$types %in% c("CONCEALED CARRY LIVENSE VIOLATION","LIQUOR LAW VIOLATION","GAMBLING","WEAPONS VIOLATION","CONCEALED CARRY LICENSE VIOLATION"), "VIOLATION", crime$types)

> crime$types <- ifelse(crime$types %in% c("HUMAN TRAFFICKING","KIDNAPPING","INTIMIDATION"), "VIOLENT CRIMES", crime$types)

> crime$types <- ifelse(crime$types %in% c("ARSON","PUBLUC PEACE VIOLATION"), "PUBLIC PEACE VIOLATION", crime$types)

> crime$types <- ifelse(crime$types %in% c("CRIMINAL DAMAGE","VIOLENT CRIMES","CRIME DAMAGE"), "CRIMINAL DAMAGE", crime$types)

> length(unique(crime$types))

[1] 17

> table(crime$types)



> #crime by types

> counts <- table(crime$types)

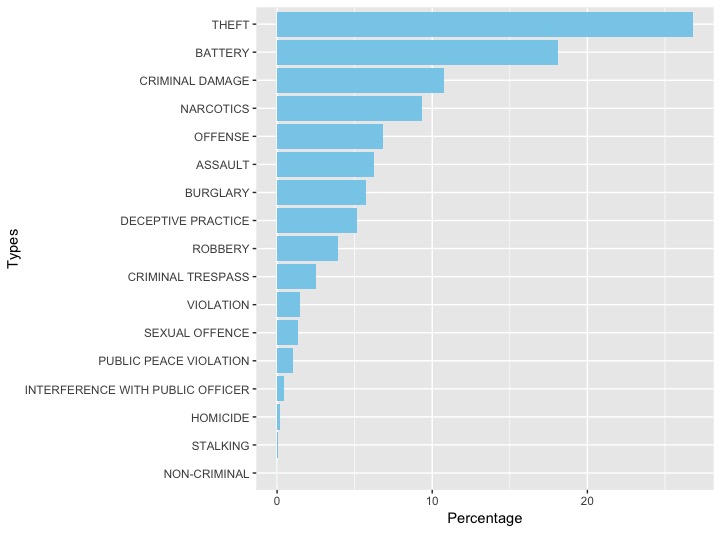
> counts = counts[order(-counts)]

> counts.percentage <- counts\*100/nrow(crime)

> counts.percentage <- as.data.frame(counts.percentage)

> ggplot(counts.percentage, aes(x=reorder(Var1, Freq), y=Freq))+

+ geom\_bar(stat="identity", fill="skyblue") + coord\_flip() +ylab("Percentage")+xlab("Types")



> #recategory location description

> length(unique(crime$Location.Description))

[1] 143

> crime$location <- as.character(crime$Location.Description)

> crime$location <- ifelse(crime$location %in% c('AIRCRAFT AIRPORT BUILDING NON-TERMINAL - NON-SECURE AREA','AIRPORT BUILDING NON-TERMINAL - SECURE AREA','AIRPORT EXTERIOR - NON-SECURE AREA','AIRPORT EXTERIOR - SECURE AREA','AIRPORT PARKING LOT','AIRPORT TERMINAL LOWER LEVEL - SECURE AREA','AIRPORT TERMINAL LOWER LEVEL - NON-SECURE AREA','AIRPORT TERMINAL MEZZANINE - NON-SECURE AREA','AIRPORT TERMINAL UPPER LEVEL - NON-SECURE AREA','AIRPORT TERMINAL UPPER LEVEL - SECURE AREA','AIRPORT TRANSPORTATION SYSTEM (ATS)','AIRPORT VENDING ESTABLISHMENT','AIRPORT/AIRCRAFT','AIRCRAFT','AIRPORT BUILDING NON-TERMINAL - NON-SECURE AREA'),'AIRPORT',crime$location)

> crime$location <- ifelse(crime$location %in% c('CTA "L" PLATFORM','CTA "L" TRAIN','CTA BUS','CTA BUS STOP','CTA GARAGE / OTHER PROPERTY','CTA PLATFORM','CTA STATION','CTA TRACKS - RIGHT OF WAY','CTA TRAIN','OTHER COMMERCIAL TRANSPORTATION','OTHER RAILROAD PROP / TRAIN DEPOT'),'CTA',crime$location)

> crime$location <- ifelse(crime$location %in% c('APPLIANCE STORE','BAR OR TAVERN','GAS STATION DRIVE/PROP.','BARBER SHOP/BEAUTY SALON','BARBERSHOP','CLEANING STORE','GARAGE/AUTO REPAIR','CLUB','CAR WASH','CLEANERS/LAUNDROMAT','NEWSSTAND','SMALL RETAIL STORE','TAVERN/LIQUOR STORE','LAUNDRY ROOM','CONVENIENCE STORE','DEPARTMENT STORE','DRUG STORE','GAS STATION','LIQUOR STORE','GROCERY FOOD STORE','PAWN SHOP','RETAIL STORE','RESTAURANT'),'STORE & SHOP',crime$location)

> crime$location <- ifelse(crime$location %in% c('ATM (AUTOMATIC TELLER MACHINE)','CURRENCY EXCHANGE','BANK','COIN OPERATED MACHINE','CREDIT UNION','SAVINGS AND LOAN'),'BANK',crime$location)

> crime$location <- ifelse(crime$location %in% c('BOAT/WATERCRAFT','AUTO','TAXI CAB','TAXICAB','VEHICLE - DELIVERY TRUCK','TRUCK','VEHICLE - OTHER RIDE SERVICE','VEHICLE NON-COMMERCIAL','RAILROAD PROPERTY','VEHICLE-COMMERCIAL','DELIVERY TRUCK'),'VEHICLE',crime$location)

> crime$location <- ifelse(crime$location %in% c('CHURCH PROPERTY','CHURCH/SYNAGOGUE/PLACE OF WORSHIP'),'CHURCH',crime$location)

> crime$location <- ifelse(crime$location %in% c('COLLEGE/UNIVERSITY GROUNDS','SCHOOL YARD','SCHOOL, PUBLIC, BUILDING','COLLEGE/UNIVERSITY RESIDENCE HALL','SCHOOL, PRIVATE, GROUNDS','SCHOOL, PRIVATE, BUILDING','SCHOOL, PUBLIC, GROUNDS','PUBLIC HIGH SCHOOL'),'SCHOOL',crime$location)

> crime$location <- ifelse(crime$location %in% c('ALLEY','BOWLING ALLEY','VACANT LOT', 'VACANT LOT/LAND' , 'STREET', 'DRIVEWAY', 'GANGWAY', 'VESTIBULE','EXPRESSWAY EMBANKMENT','HIGHWAY/EXPRESSWAY','SIDEWALK','PARK PROPERTY','LAKEFRONT/WATERFRONT/RIVERBANK','CONSTRUCTION SITE','BRIDGE','LAGOON','CEMETARY','PARKING LOT/GARAGE(NON.RESID.)','FOREST PRESERVE','YARD'), 'STREET', crime$location)

> crime$location <- ifelse(crime$location %in% c('ANIMAL HOSPITAL','NURSING HOME', 'NURSING HOME/RETIREMENT HOME','DAY CARE CENTER', 'HOSPITAL BUILDING/GROUNDS', 'HOSPITAL', 'MEDICAL/DENTAL OFFICE' ), 'HOSPITAL', crime$location)

> crime$location <- ifelse(crime$location %in% c('APARTMENT','DRIVEWAY - RESIDENTIAL','GARAGE','HALLWAY','HOUSE','RESIDENCE','RESIDENCE-GARAGE','RESIDENTIAL YARD (FRONT/BACK)','BASEMENT','STAIRWELL','CHA APARTMENT','CHA HALLWAY/STAIRWELL/ELEVATOR','CHA PARKING LOT','CHA PARKING LOT/GROUNDS','ELEVATOR','PARKING LOT','PORCH'), 'RESIDENCE', crime$location)

> crime$location <- ifelse(crime$location %in% c('FACTORY/MANUFACTURING BUILDING','FEDERAL BUILDING','FIRE STATION', 'GOVERNMENT BUILDING','POLICE FACILITY/VEH PARKING LOT', 'GOVERNMENT BUILDING/PROPERTY' ,'JAIL / LOCK-UP FACILITY'),'GOVERMENT', crime$location)

> crime$location <- ifelse(crime$location %in% c('ABANDONED BUILDING','ATHLETIC CLUB','COMMERCIAL / BUSINESS OFFICE','MOVIE HOUSE/THEATER','OFFICE','LIBRARY','POOL ROOM','POOLROOM','SPORTS ARENA/STADIUM','WAREHOUSE'),'PUBLIC BUILDING', crime$location)

> crime$location <- ifelse(crime$location %in% c('TAVERN','HOTEL','MOTEL','HOTEL/MOTEL'),'HOTEL',crime$location)

> length(unique(crime$location))

[1] 16

> #location analysis

> location <- table(crime$location)

> location = location[order(-location)]

> location.percentage <- location\*100/nrow(crime)

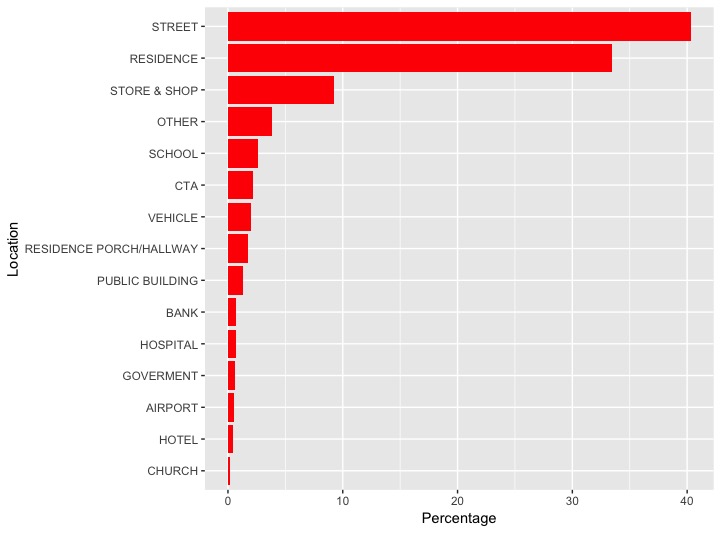
> location.percentage



> location.percentage <- as.data.frame(location.percentage)

> ggplot(location.percentage[1:15,], aes(x=reorder(Var1, Freq), y=Freq))+

+ geom\_bar(stat="identity", fill="red") + coord\_flip() +ylab("Percentage")+xlab("Location")



# IV. Data Mining Techniques and Implementation

overall time series trend

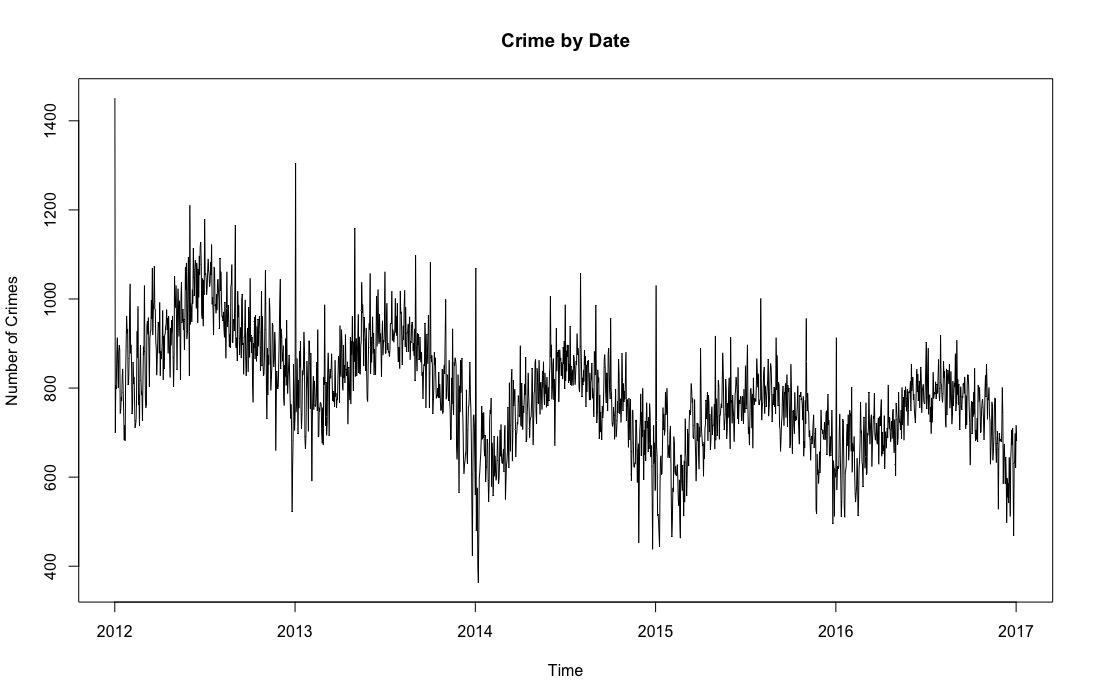
> #ts

> crime\_by\_date <- table(crime$Date)

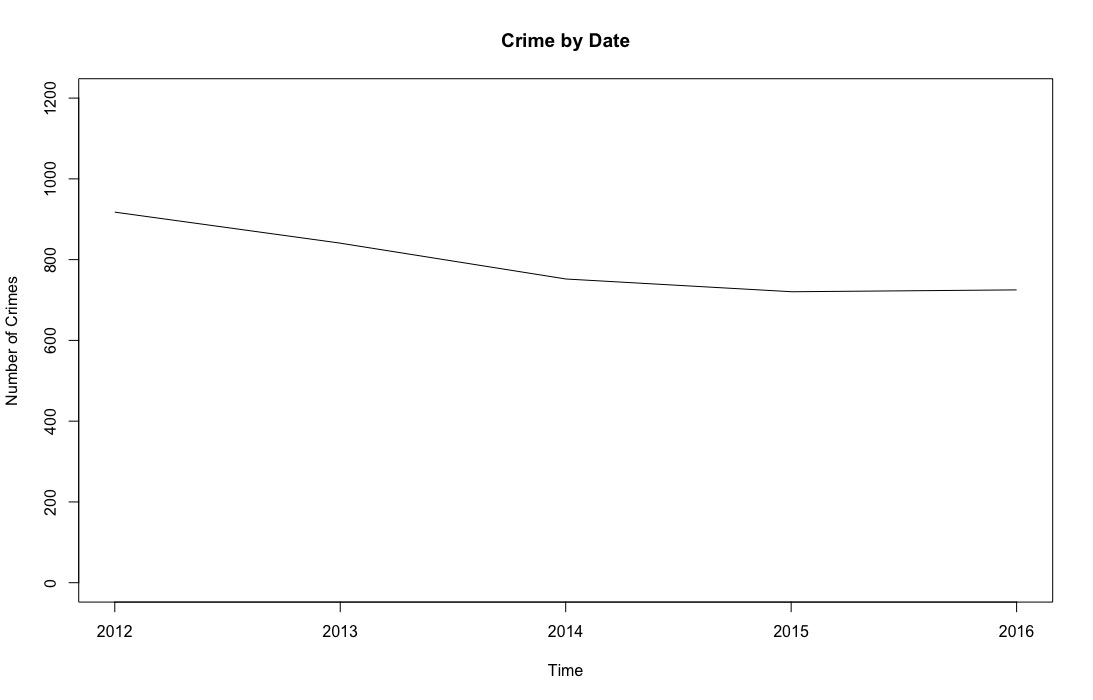
> crime\_by\_date <- as.data.frame(crime\_by\_date)

> crime\_by\_date\_ts <- ts(crime\_by\_date$Freq, start = c(2012,1), freq=365)

> plot(crime\_by\_date\_ts, xlab= 'Time', ylab= 'Number of Crimes', main="Crime by Date")



> plot(aggregate(crime\_by\_date\_ts, FUN = mean), xlab='Time', ylab='Number of Crimes', ylim=c(0,1200), main="Crime by Date")

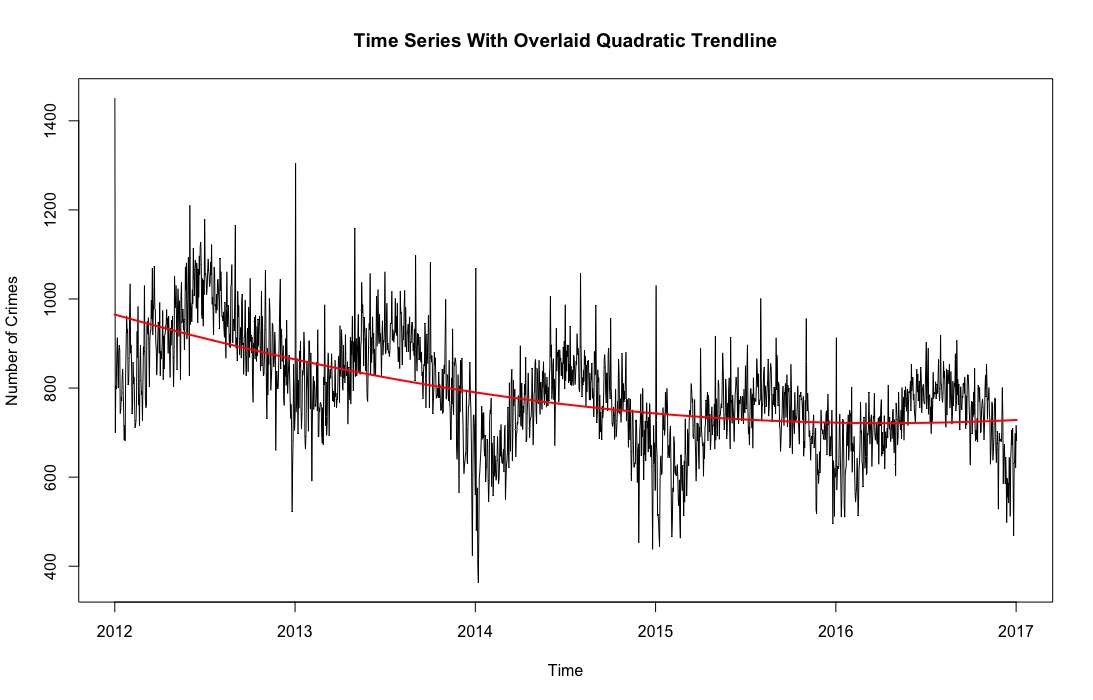


> ####forecast

> crime\_by\_date\_lm <- tslm(crime\_by\_date\_ts ~ trend + I(trend^2))

> plot(crime\_by\_date\_ts, xlab="Time", ylab= "Number of Crimes", main="Time Series With Overlaid Quadratic Trendline")

> lines(crime\_by\_date\_lm$fitted.values, lwd=2, col='red', main="Crime by Date")#overlay the fitted values of the linear model



> ####navie and seasonal naive forecasets

> crime\_valid <- 366

> crime\_train <- length(crime\_by\_date\_ts) - crime\_valid

> train\_ts <- window(crime\_by\_date\_ts, start=c(2012,1), end=c(2012, crime\_train))

> valid\_ts <- window(crime\_by\_date\_ts, start=c(2012, crime\_train+1), end=c(2012, crime\_train+crime\_valid))

> #generate the naive and seasonal naive forecasts

> naive.pred <- naive(train\_ts, h=crime\_valid)

> snaive.pred <- snaive(train\_ts, h=crime\_valid)

> #plot forecasts and actuals in the training and validation sets

> plot(train\_ts, ylab="Number of Crime", xlab="Time", xlim=c(2012,2019), ylim=c(300,1800), main="Naive & Seasonal Naive Forecasts")

> axis(1, at = seq(2012, 2017,1), labels = format(seq(2012,2017,1)))

> lines(naive.pred$mean, lwd=2, col="skyblue3", lty=3)

> lines(snaive.pred$mean, lwd=2, col="skyblue3", lty=3)

> lines(valid\_ts, col="darkorchid4", lty=4)

> lines(c(2017-1,2017-1), c(0,3500))

> lines(c(2017,2017), c(0,3500))

> text(2014, 1700,"Training")

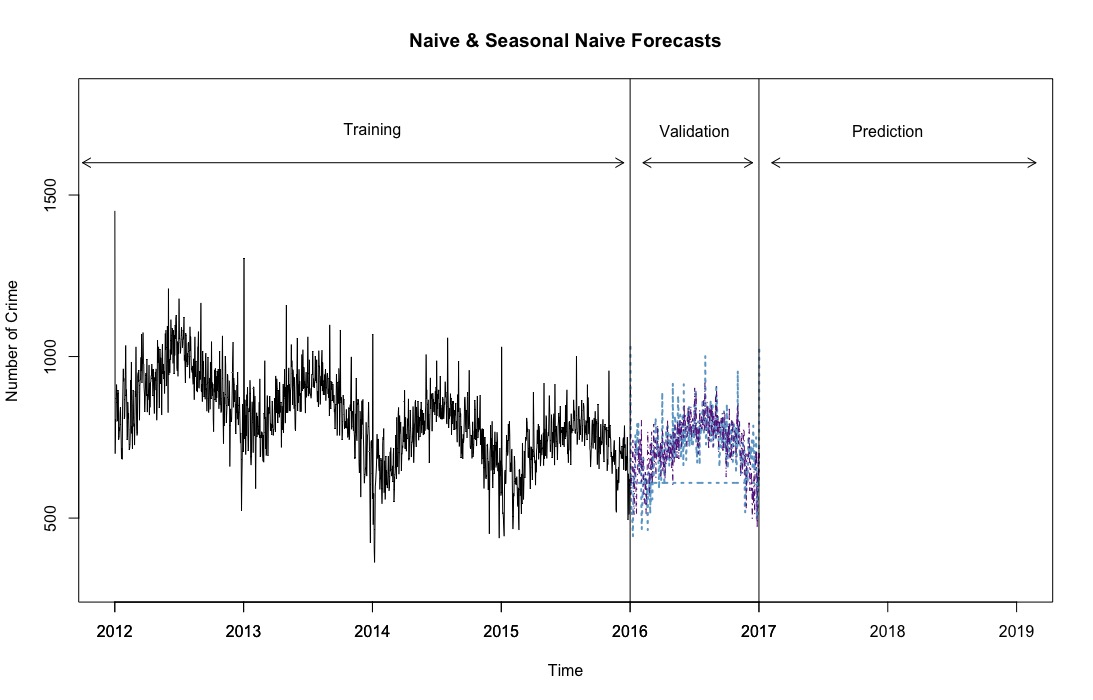
> text(2016.5, 1700,"Validation")

> text(2018, 1700,"Prediction")

> arrows(2015.95, 1600, 2012.75-1, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2016.1, 1600, 2016.95, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2017.1, 1600, 2019.15, code = 3, length = 0.1, lwd = 1, angle = 30)



> #predictive accuracy

> accuracy(naive.pred, valid\_ts)



> accuracy(snaive.pred, valid\_ts)



> ####ts by month

> crime\_by\_month <- table(crime$Month, crime$Year)

> crime\_by\_month <- as.data.frame(crime\_by\_month)

> crime\_by\_month\_ts <- ts(crime\_by\_month$Freq, start = c(2012,1), frequency = 12)

> plot(crime\_by\_month\_ts, xlab="Time", ylab="Number of Crimes", ylim = c(10000,40000), main = "Crime by Month")



> ####naive and seasonal naive forecasts in 1 year validation set

> crime\_valid1 <- 12

> crime\_train1 <- length(crime\_by\_month\_ts) - crime\_valid1

> train\_ts1 <- window(crime\_by\_month\_ts, start=c(2012,1), end=c(2012, crime\_train1))

> valid\_ts1 <- window(crime\_by\_month\_ts, start=c(2012, crime\_train1+1), end=c(2012, crime\_train1+crime\_valid1))

> #generate the naive and seasonal naive forecasts

> naive.pred1 <- naive(train\_ts1, h=crime\_valid1)

> snaive.pred1 <- snaive(train\_ts1, h=crime\_valid1)

> #plot forecasts and actuals in the training and validation sets

> plot(train\_ts1, ylab="Number of Crime", xlab="Time", xlim=c(2012,2019), ylim=c(10000,40000), main="Naive & Seasonal Naive Forecasts")

> axis(1, at = seq(2012, 2017,1), labels = format(seq(2012,2017,1)))

> lines(naive.pred1$mean, lwd=2, col="blue", lty=3)

> lines(snaive.pred1$mean, lwd=2, col="blue", lty=3)

> lines(valid\_ts1, col="red", lty=5)

> lines(c(2017-1,2017-1), c(0,45000))

> lines(c(2017,2017), c(0,45000))

> text(2014, 37500,"Training")

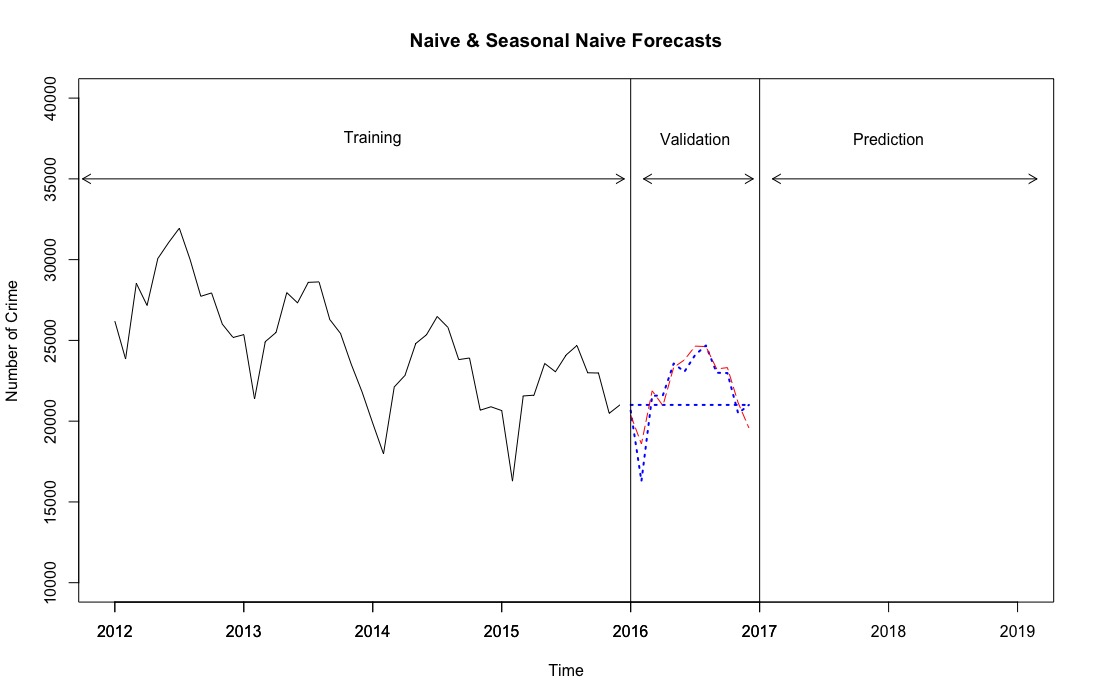
> text(2016.5, 37500,"Validation")

> text(2018, 37500,"Prediction")

> arrows(2015.95, 35000, 2012.75-1, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2016.1, 35000, 2016.95, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2017.1, 35000, 2019.15, code = 3, length = 0.1, lwd = 1, angle = 30)



> ####regression-based forecasting

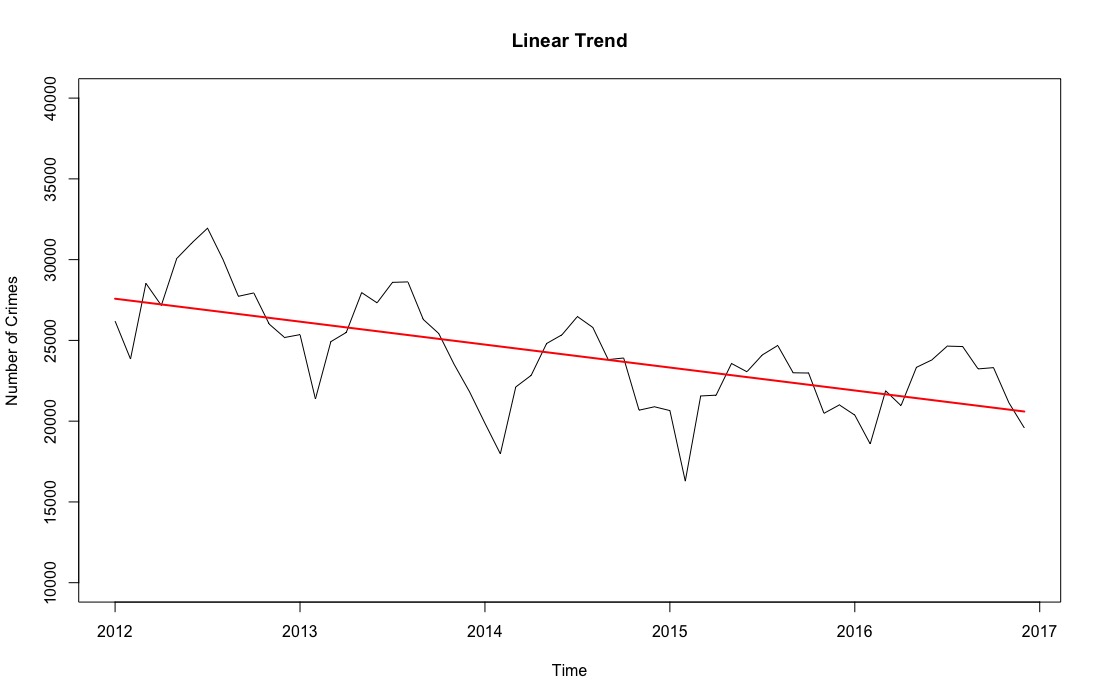
> #linear trend

> library(forecast)

> crime\_by\_month\_lm <- tslm (crime\_by\_month\_ts ~ trend)

> plot(crime\_by\_month\_ts, xlab="Time", ylab="Number of Crimes", ylim=c(10000, 40000), main="Linear Trend")

> lines(crime\_by\_month\_lm$fitted.values, lwd =2, col="red")



> #linear trend + seasonal

> train.lm.trend.season <- tslm(train\_ts1 ~ trend + I(trend^2) + season)

> train.lm.trend.season.pred <- forecast(train.lm.trend.season, h = crime\_valid1, level = 0)

> par(mfrow = c(1,1))

> plot(train.lm.trend.season.pred, ylab="Number of Crime", xlab="Time", xlim=c(2012,2019), ylim=c(10000,50000), main="Regression Model With Seasonality Applied")

> axis(1, at = seq(2012, 2017,1), labels = format(seq(2012,2017,1)))

> lines(train.lm.trend.season.pred$fitted, lwd=2, col="skyblue")

> lines(valid\_ts1, col = "Red")

> lines(c(2017-1,2017-1), c(0,50000))

> lines(c(2017,2017), c(0,50000))

> text(2014, 47500,"Training")

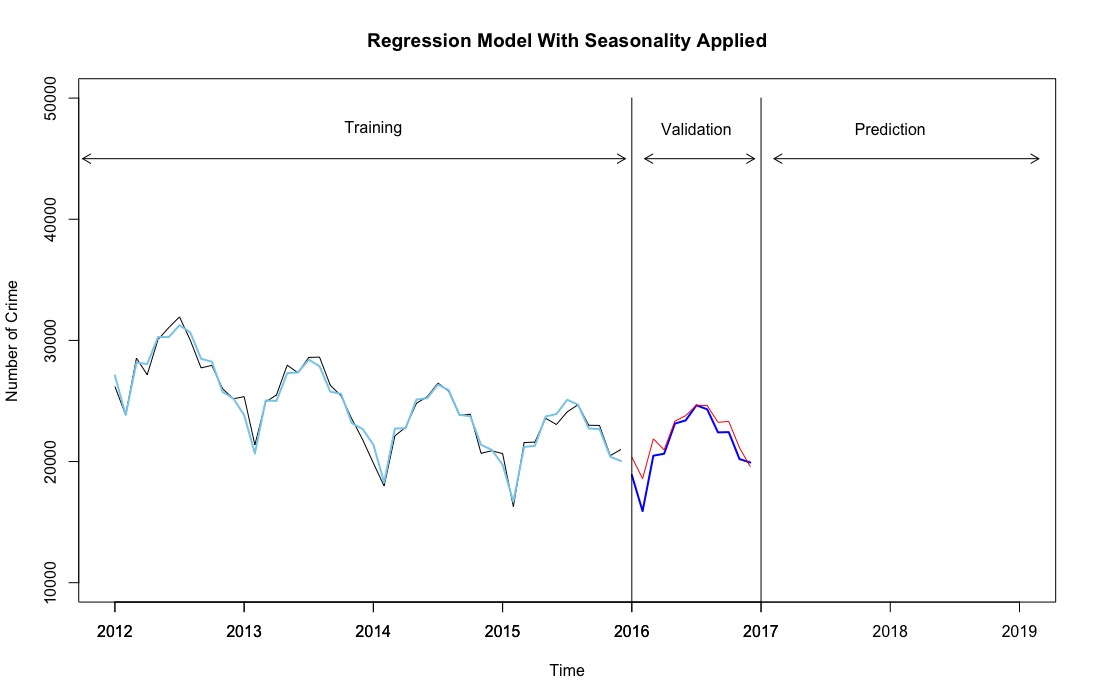
> text(2016.5, 47500,"Validation")

> text(2018, 47500,"Prediction")

> arrows(2015.95, 45000, 2012.75-1, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2016.1, 45000, 2016.95, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2017.1, 45000, 2019.15, code = 3, length = 0.1, lwd = 1, angle = 30)



> plot(train.lm.trend.season.pred$residuals, ylab= "Forecast Errors", xlab= "Time", xlim=c(2012,2019), ylim=c(-4000, 5500))

> axis(1, at = seq(2012, 2017,1), labels = format(seq(2012,2017,1)))

> lines(valid\_ts1 - train.lm.trend.season.pred$mean)

> lines(c(2017-1,2017-1), c(-7000, 12000))

> lines(c(2017,2017), c(-7000, 12000))

> text(2014, 5000,"Training")

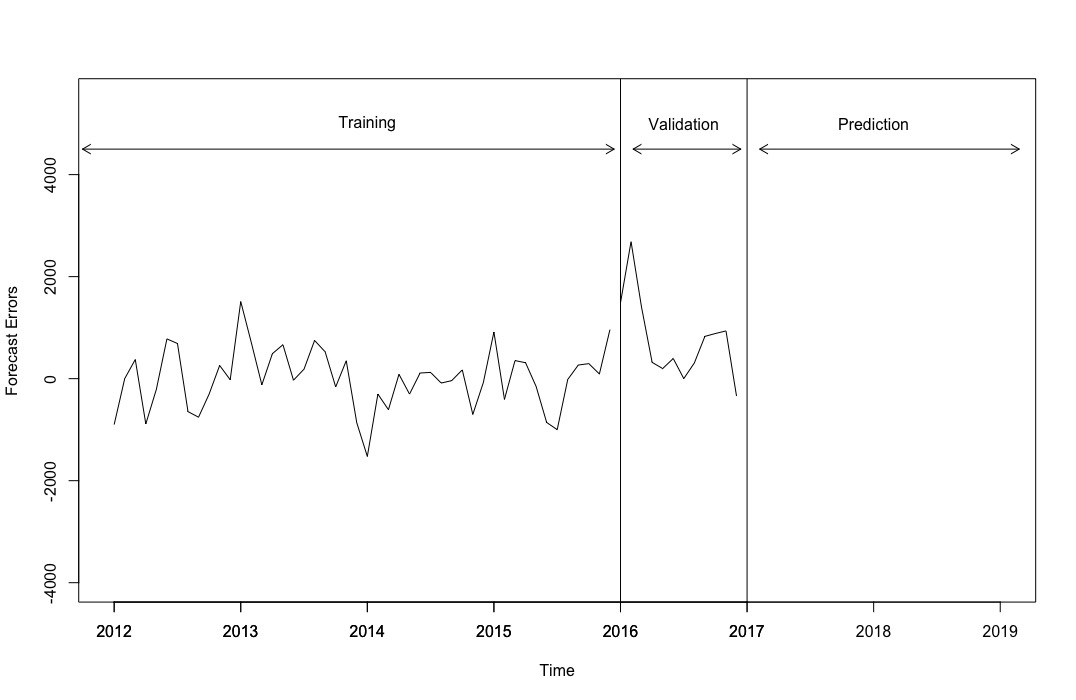
> text(2016.5, 5000,"Validation")

> text(2018, 5000,"Prediction")

> arrows(2015.95, 4500, 2012.75-1, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2016.1, 4500, 2016.95, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2017.1, 4500, 2019.15, code = 3, length = 0.1, lwd = 1, angle = 30)



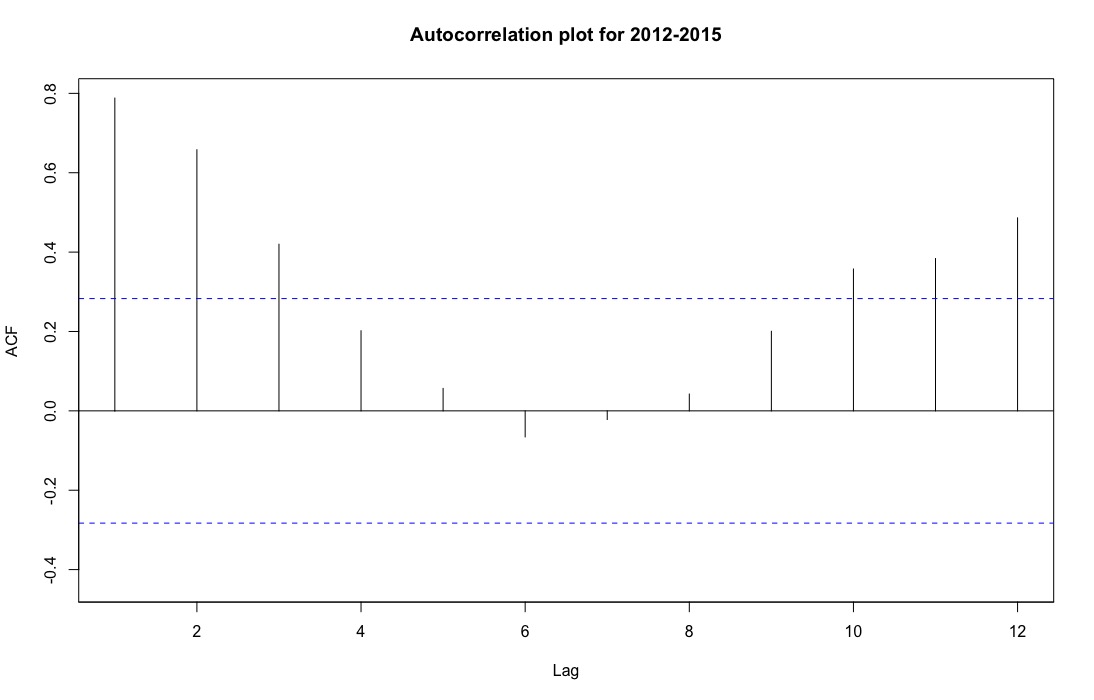
> summary(train.lm.trend.season)



> #computing autocorrelation plot for lags 1-12

> crime\_48.ts <- window(train\_ts1, start=c(2012,1), end=c(2015,12))

> Acf(crime\_48.ts, lag.max = 12, main="Autocorrelation plot for 2012-2015")



> #ARIMA

> train.res.arima <- Arima(train.lm.trend.season$residuals, order = c(1, 0, 0))

> valid.res.arima.pred <- forecast(train.res.arima, h = 1)

> summary(train.res.arima)



> valid.res.arima.pred



> #residuals

> plot(train.lm.trend.season$residuals, ylab = "Residuals", xlab= "Time", xlim=c(2012,2019), ylim=c(-2000, 2000), main=" AR(1) Model To The Residual Series")

> lines(valid.res.arima.pred$residuals, lwd = 2, col="red")

> lines(c(2017-1,2017-1), c(-7000, 12000))

> lines(c(2017,2017), c(-7000, 12000))

> text(2014, 2000,"Training")

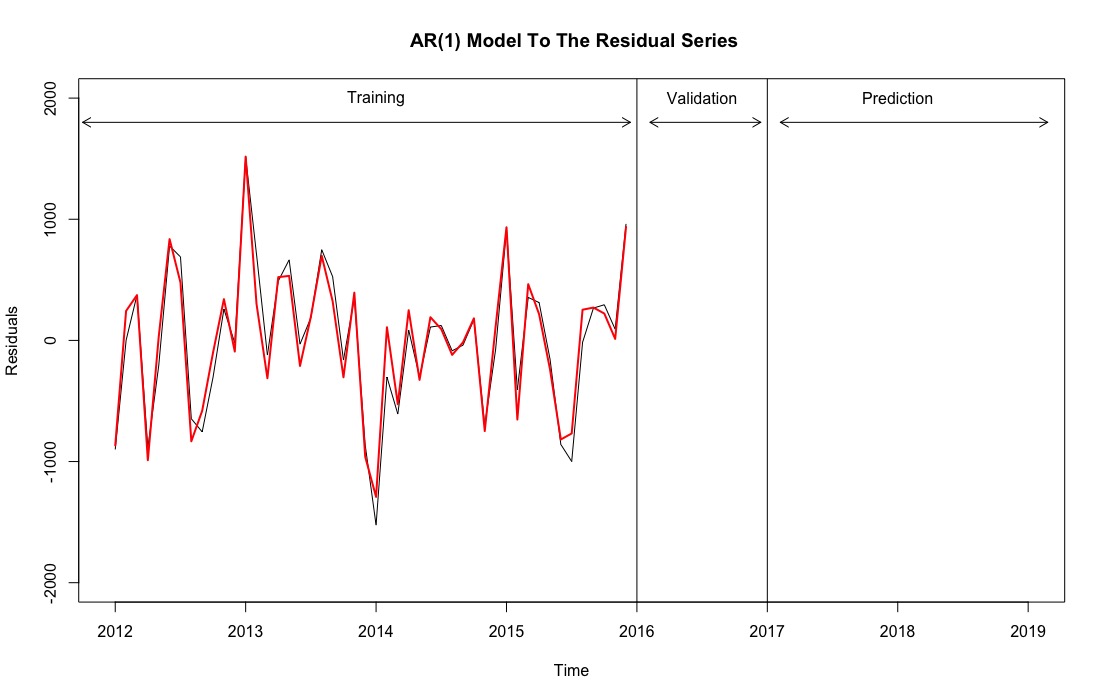
> text(2016.5, 2000,"Validation")

> text(2018, 2000,"Prediction")

> arrows(2015.95, 1800, 2012.75-1, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2016.1, 1800, 2016.95, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2017.1, 1800, 2019.15, code = 3, length = 0.1, lwd = 1, angle = 30)



> #output for simple exponential smoothing forecaster with alpha=0.2, applied to the series of residualsfrom the regression model

> residuals.ts <- train.lm.trend.season$residuals

> ses <- ets(residuals.ts, model = "ANN", alpha = 0.9)

> ses.pred <- forecast(ses, h= crime\_valid1, level = 0)

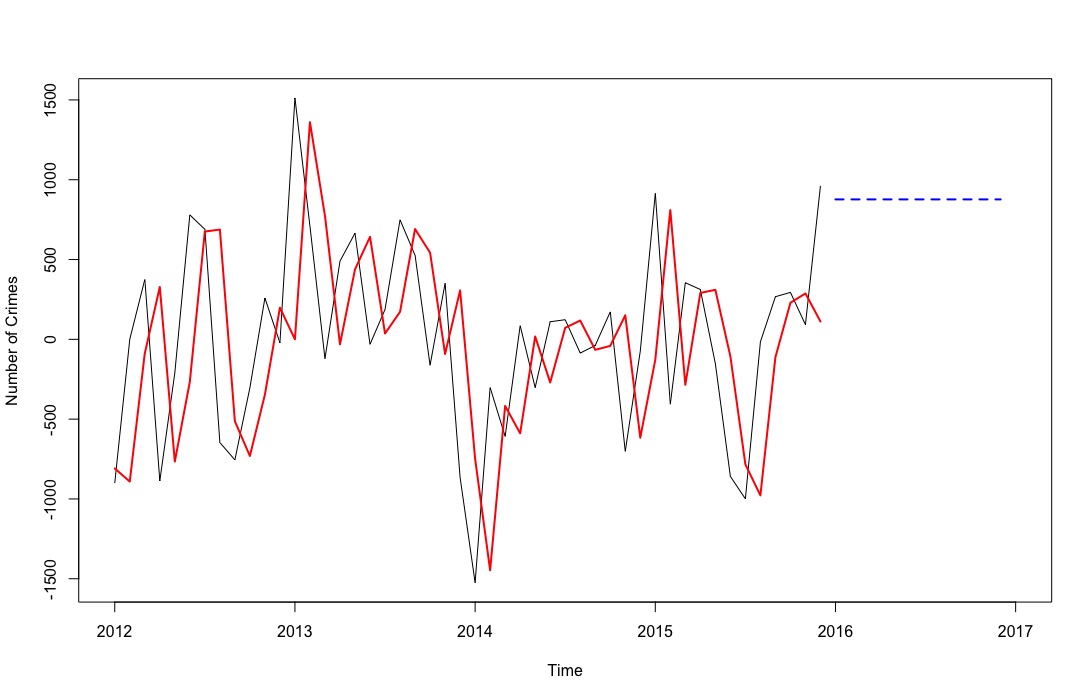
> plot(ses.pred, ylab = "Number of Crimes", xlab = "Time", xlim=c(2012,2017), main = "", flty =2)

> train.lm.trend.season.pred <- forecast(train.lm.trend.season, h=crime\_valid1, level = 1)

> lines(train.lm.trend.season.pred$fitted, lwd=1, col="blue")

> lines(ses.pred$fitted, lwd=2, col="red")

> lines(valid\_ts1)



> #Holt-Winters exponential smoothing

> hwin <- ets(train\_ts1, model = "MAA")

> hwin.pred <- forecast(hwin, h = crime\_valid1, level = 0)

> # plot the series

> plot(hwin.pred, ylim = c(10000, 40000), ylab = "Number of Crimes", xlab = "Time",

+ bty = "l", xaxt = "n", xlim = c(2012,2019), main = "Holt-Winters exponential smoothing", flty = 2)

> axis(1, at = seq(2012, 2019, 1), labels = format(seq(2012, 2019, 1)))

> lines(hwin.pred$fitted, lwd = 2, col = "blue")

> lines(valid\_ts1)

> lines(c(2017-1,2017-1), c(0, 50000))

> lines(c(2017,2017), c(0, 50000))

> text(2014, 37500,"Training")

> text(2016.5, 37500,"Validation")

> text(2018, 37500,"Prediction")

> arrows(2015.95, 35000, 2012.75-1, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2016.1, 35000, 2016.95, code = 3, length = 0.1, lwd = 1, angle = 30)

> arrows(2017.1, 35000, 2019.15, code = 3, length = 0.1, lwd = 1, angle = 30)