Employee Absenteeism

Problem statement:

XYZ is a courier company. As we appreciate that human capital plays an important role in collection, transportation and delivery. The company is passing through genuine issue of Absenteeism.
The company has shared it dataset and requested to have an answer on the following areas:

1. What changes company should bring to reduce the number of absenteeism?
2. How much losses every month can we project in 2011 if same trend of absenteeism continues?

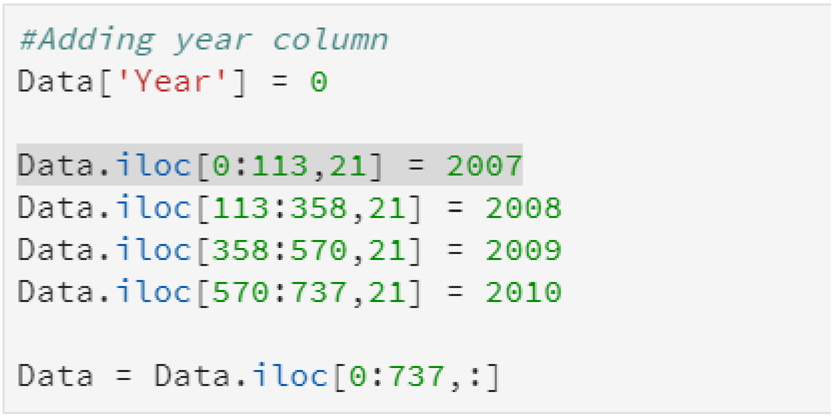
Build suitable model (both R and Python) to answer the above two questions with a proper report .

Dataset Details:
Dataset Characteristics: Timeseries Multivariant
Number of Attributes: 21
Missing Values : Yes

Considerations:

As per problem statement, Forecasting has to be done for the year 2011. In the dataset, year has not been explicitly said. As it is a forecasting problem, We are considering that data is from 2007 to 2010.

Year column has been added to the dataset which represents above statement.



```
#Adding year column
Data['Year'] = 0

Data.iloc[0:113,21] = 2007
Data.iloc[113:358,21] = 2008
Data.iloc[358:570,21] = 2009
Data.iloc[570:737,21] = 2010

Data = Data.iloc[0:737,:]
```

For the consideration, we have data from starting month 7 of 2007 to 7 of 2010.
Projection will be ahead of after 7 of 2010

Last three observation has no month and absenteeism data, so we neglect them and now we have total of 737 observations

Basic exploration:

Let's perform some basic data analysis, which will help us to find the statistical properties of the training dataset. This kind of analysis is called exploratory data analysis and it will help us understand how our dataset represents the facts. After deriving some facts, we can use them in order to derive feature engineering.

Basic data analysis is a crucial step for understanding data. Now on wards, you will encounter code in R or Python with output explanation. All codes will be attached in attachments and some referred here too.
*Note 1 : Accuracy and other metrics measure has been posted with respect to R code output. Python code accuracy may vary.

Steps of data analysis:

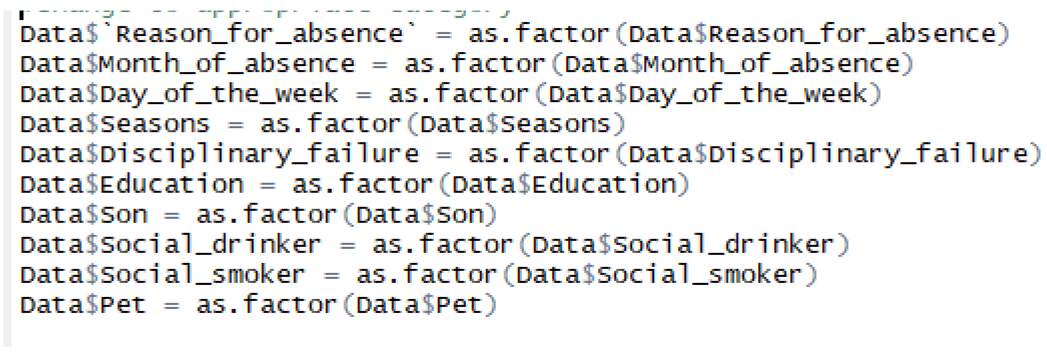
1. Convert variables into required datatype
2. Listing statistical properties
3. Missing value analysis
4. Outlier analysis
5. Feature selection
6. Feature scaling

These analysis on data will bring us tremendous value while modeling data. Understanding data on this manner gives way for us to better tune our parameters which in turn improve our accuracy. After processing data model selection and finalization process can be done.

We will walk through all the steps of data analysis above mentioned and pause in Outlier analysis. Before dealing outlier, First question in the problem statement will be answered by deriving facts. Then, we will continue with outlier analysis up-to building model.

1. Convert variables into required datatype :

All variables are in num format. Some variables should be in factor format according to their values. So,



```
Data$`Reason_for_absence` = as.factor(Data$Reason_for_absence)
Data$Month_of_absence = as.factor(Data$Month_of_absence)
Data$Day_of_the_week = as.factor(Data$Day_of_the_week)
Data$Seasons = as.factor(Data$Seasons)
Data$Disciplinary_failure = as.factor(Data$Disciplinary_failure)
Data$Education = as.factor(Data$Education)
Data$Son = as.factor(Data$Son)
Data$Social_drinker = as.factor(Data$Social_drinker)
Data$Social_smoker = as.factor(Data$Social_smoker)
Data$Pet = as.factor(Data$Pet)
```

These variables are considered as Categorical variable for this dataset.

2. Listing statistical properties

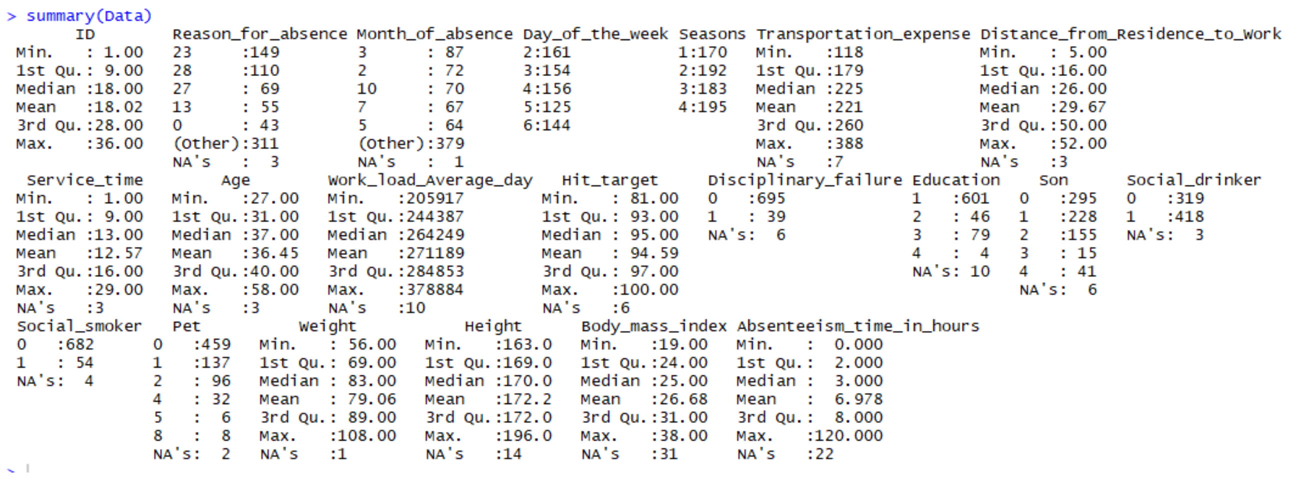
In this section, we will get an idea about the statistical properties of the training dataset. Using summary function in R, we can list out the following properties:

For continuous variable:

1. Min: Minimum value of the given column
2. 1st Quartile: 25th percentile of the given column
3. Median: 50th percentile or center value after ordering dataset
4. Mean: Average value of the given dataset
5. 3rd Quartile: 75th percentile of the given column
6. Max: Maximum value of the given column

For categorical variable:

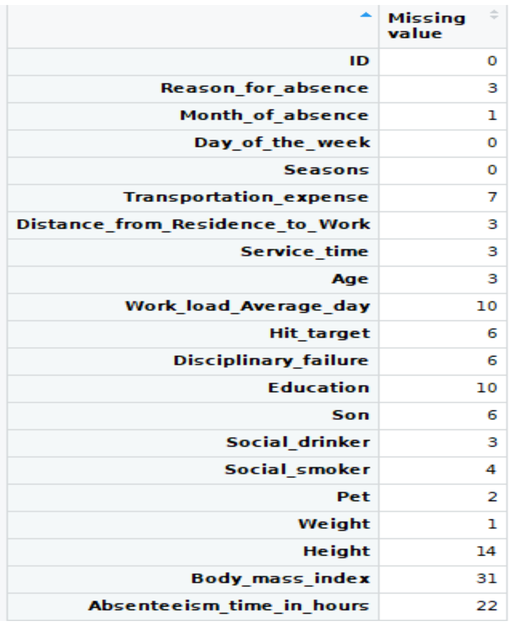
1. Count: Count of observations in the each category of a given column



```
> summary(Data)
      ID        Reason_for_absence Month_of_absence Day_of_the_week Seasons Transportation_expense Distance_from_Residence_to_Work
Min.   : 1.00   23     :149        3      : 87      2:161           1:170   Min.   :118            Min.   : 5.00
1st Qu.: 9.00   28     :110        2      : 72      3:154           2:192   1st Qu.:179            1st Qu.:16.00
Median :18.00   27     : 69        10     : 70      4:156           3:183   Median :225            Median :26.00
Mean   :18.02   13     : 55        7      : 67      5:125           4:195   Mean   :221            Mean   :29.67
3rd Qu.:28.00   0      : 43        5      : 64      6:144                   3rd Qu.:260            3rd Qu.:50.00
Max.   :36.00   (Other):311        (Other):379                              Max.   :388            Max.   :52.00
                NA's   :  3        NA's   :  1                              NA's   :7              NA's   :3
 Service_time        Age        Work_load_Average_day   Hit_target      Disciplinary_failure Education    Son      Social_drinker
Min.   : 1.00   Min.   :27.00   Min.   :205917        Min.   : 81.00    0   :695             1   :601   0   :295   0   :319
1st Qu.: 9.00   1st Qu.:31.00   1st Qu.:244387        1st Qu.: 93.00    1   : 39             2   : 46   1   :228   1   :418
Median :13.00   Median :37.00   Median :264249        Median : 95.00    NA's:  6             3   : 79   2   :155   NA's:  3
Mean   :12.57   Mean   :36.45   Mean   :271189        Mean   : 94.59                         4   :  4   3   : 15
3rd Qu.:16.00   3rd Qu.:40.00   3rd Qu.:284853        3rd Qu.: 97.00                         NA's: 10   4   : 41
Max.   :29.00   Max.   :58.00   Max.   :378884        Max.   :100.00                                    NA's:  6
NA's   :3       NA's   :3       NA's   :10            NA's   :6
Social_smoker   Pet          Weight           Height      Body_mass_index Absenteeism_time_in_hours
0   :682      0   :459   Min.   : 56.00   Min.   :163.0   Min.   :19.00   Min.   :  0.000
1   : 54      1   :137   1st Qu.: 69.00   1st Qu.:169.0   1st Qu.:24.00   1st Qu.:  2.000
NA's:  4      2   : 96   Median : 83.00   Median :170.0   Median :25.00   Median :  3.000
              4   : 32   Mean   : 79.06   Mean   :172.2   Mean   :26.68   Mean   :  6.978
              5   :  6   3rd Qu.: 89.00   3rd Qu.:172.0   3rd Qu.:31.00   3rd Qu.:  8.000
              8   :  8   Max.   :108.00   Max.   :196.0   Max.   :38.00   Max.   :120.000
              NA's:  2   NA's   :1        NA's   :14      NA's   :31      NA's   :22
```

3. Missing value Analysis

In order to find the missing values in the dataset, we need to check each and every column in the dataset. If we found any, we can replace with more meaningful values.



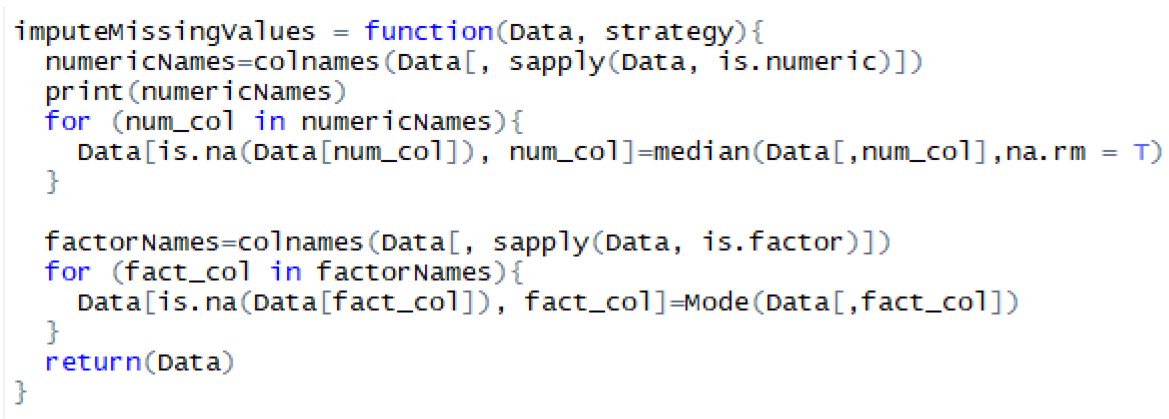
	Missing value
ID	0
Reason_for_absence	3
Month_of_absence	1
Day_of_the_week	0
Seasons	0
Transportation_expense	7
Distance_from_Residence_to_Work	3
Service_time	3
Age	3
Work_load_Average_day	10
Hit_target	6
Disciplinary_failure	6
Education	10
Son	6
Social_drinker	3
Social_smoker	4
Pet	2
Weight	1
Height	14
Body_mass_index	31
Absenteeism_time_in_hours	22

Almost all the variables has missing values. Impute those numerical missing values by one of the below:

1. Median value
2. Mean value
3. With distance measures(KNN)

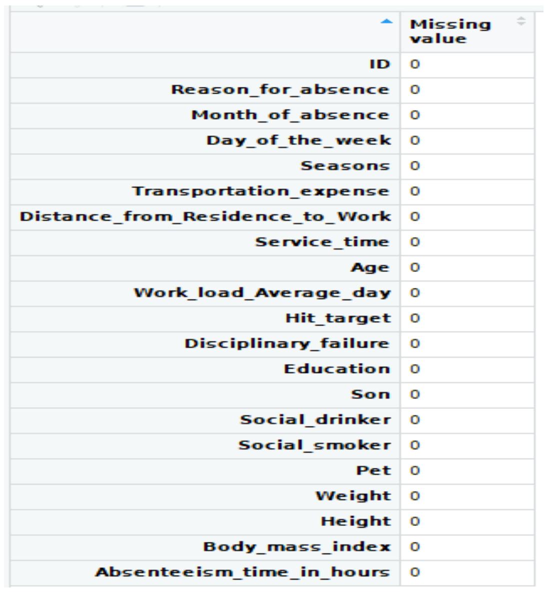
Here, Median value replacement suits better than another methods for numerical types.

For categorical variables, missing values will be replaced with mode.



```r
imputeMissingValues = function(Data, strategy){
  numericNames=colnames(Data[, sapply(Data, is.numeric)])
  print(numericNames)
  for (num_col in numericNames){
    Data[is.na(Data[num_col]), num_col]=median(Data[,num_col],na.rm = T)
  }

  factorNames=colnames(Data[, sapply(Data, is.factor)])
  for (fact_col in factorNames){
    Data[is.na(Data[fact_col]), fact_col]=Mode(Data[,fact_col])
  }
  return(Data)
}
```

After imputation,



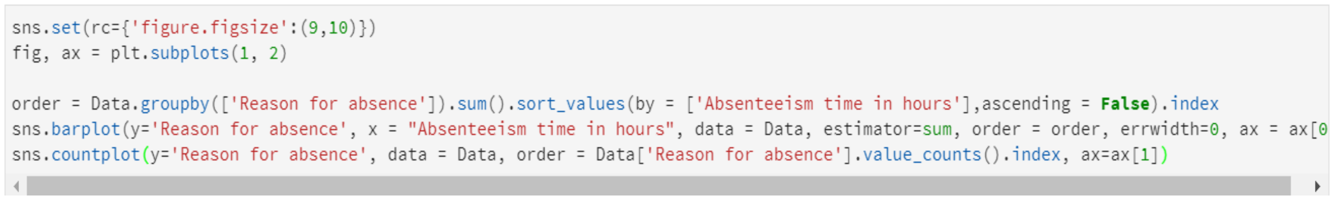
	Missing value
ID	0
Reason_for_absence	0
Month_of_absence	0
Day_of_the_week	0
Seasons	0
Transportation_expense	0
Distance_from_Residence_to_Work	0
Service_time	0
Age	0
Work_load_Average_day	0
Hit_target	0
Disciplinary_failure	0
Education	0
Son	0
Social_drinker	0
Social_smoker	0
Pet	0
Weight	0
Height	0
Body_mass_index	0
Absenteeism_time_in_hours	0

All missing values has been treated with meaningful values.

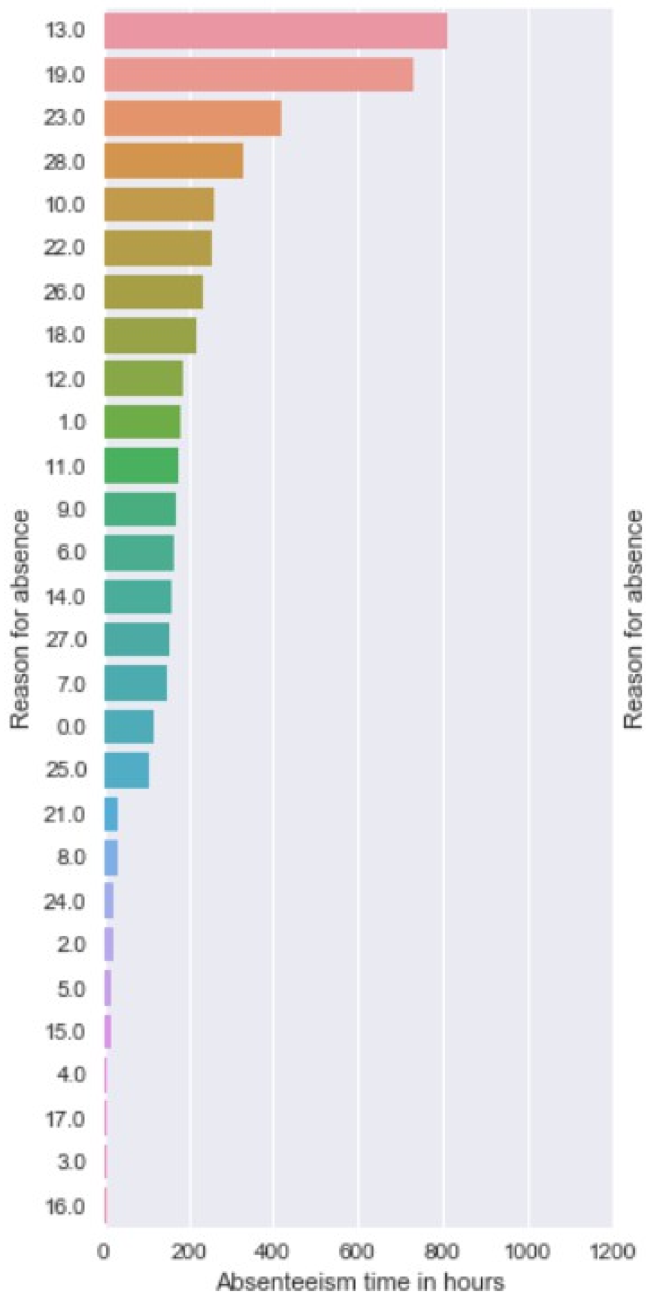
As I already stated above, Now we are going to derive the facts from the dataset and formalize facts as policies for company which help them to decrease the absenteeism of the employees.

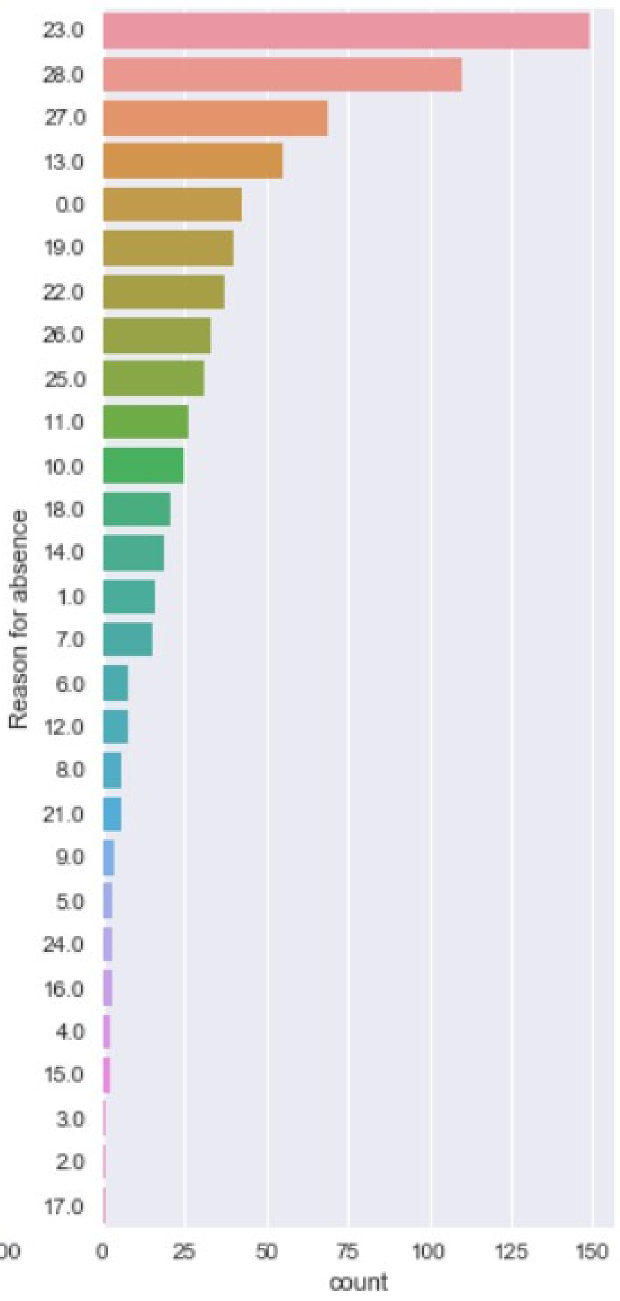
Changes company should make in order to reduce absenteeism.

1. Most employee's absenteeism caused due to the Medical consultation and dental consultation. Take a look at below graph.



```python
sns.set(rc={'figure.figsize':(9,10)})
fig, ax = plt.subplots(1, 2)

order = Data.groupby(['Reason for absence']).sum().sort_values(by = ['Absenteeism time in hours'],ascending = False).index
sns.barplot(y='Reason for absence', x = "Absenteeism time in hours", data = Data, estimator=sum, order = order, errwidth=0, ax = ax[0
sns.countplot(y='Reason for absence', data = Data, order = Data['Reason for absence'].value_counts().index, ax=ax[1])
```





Even though largest absent hours is due to Reason 13 and 19, we should also taken into considerations of absenteeism count with respect to Reason(Second graph) because it generalizes and collects the reason among the employees.

For example., Let's map the number 13 with its reason. It is

“Diseases of the musculoskeletal system and connective tissue ”

Diseases of these kind lead to long absence but don't occur for many. That's why we are also taking “count” for business decision.

But if we took 23 which means “Medical consultation”, then it occurs widely among all employees but lead to considerably short when compared to 13.

Hence Each graph demanding different changes. Now, we can infer from the above graph for the reason of absence.

For overcoming this kind of absenteeism,

“Office Eco-system and architecture should be changed in such a way that employee health as prior concern”

Because skeletal and connective tissue problems may be occurring due to office habits like sitting posture, long sitting hours etc. So, its good to change the office architecture employee's health centric.

“Medical campaign should conducted from the company for employees once in a year”

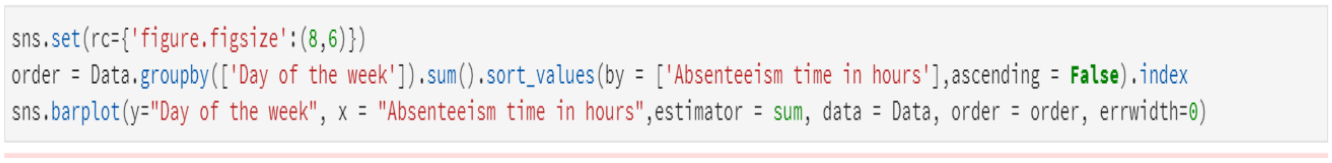
Because most of the employee's took leave for number 23 and 28 which is nothing but a Medical and Dental consultation. If company itself run a campaign once in a year, then it will reduce absenteeism drastically.

The above decisions reduce employee absenteeism considerably and increases respect towards company.

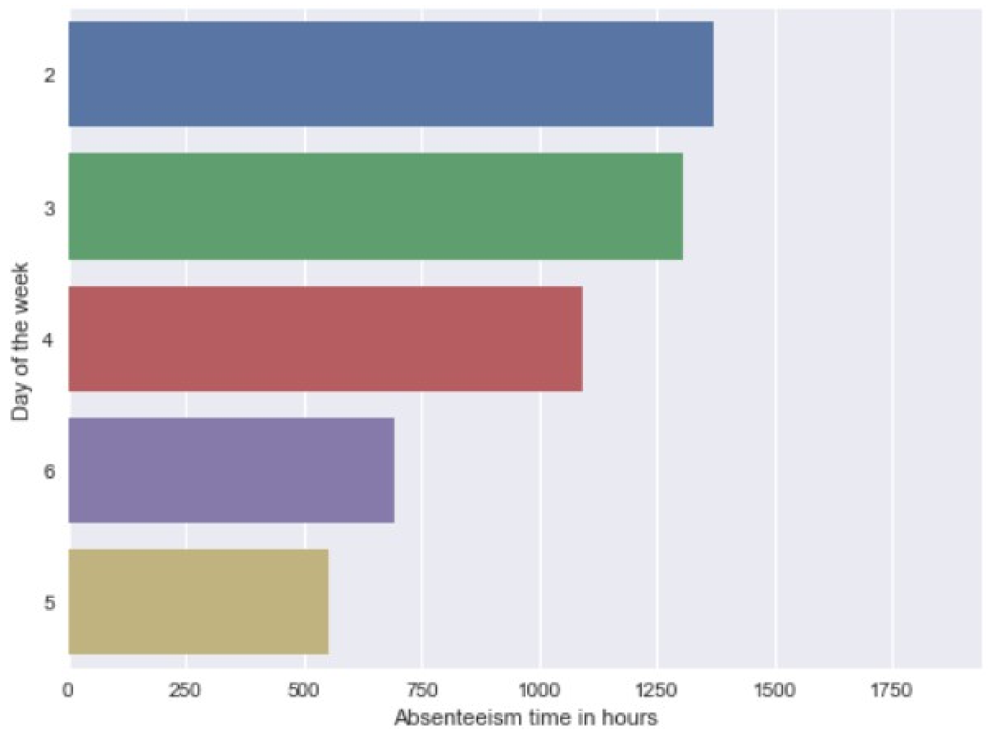
2. **Lack of motivation.**

Absenteeism has been high in Monday and reduces way to the Friday. This clearly happens because of uninterested work or lack of Monday motivation etc.

“Conduct some special minor fun events , reduce service time on mondays, display motivational quotes etc. Ultimately make your employee happy and help them stay motivated”



```
sns.set(rc={'figure.figsize':(8,6)})
order = Data.groupby(['Day of the week']).sum().sort_values(by = ['Absenteeism time in hours'],ascending = False).index
sns.barplot(y="Day of the week", x = "Absenteeism time in hours",estimator = sum, data = Data, order = order, errwidth=0)
```



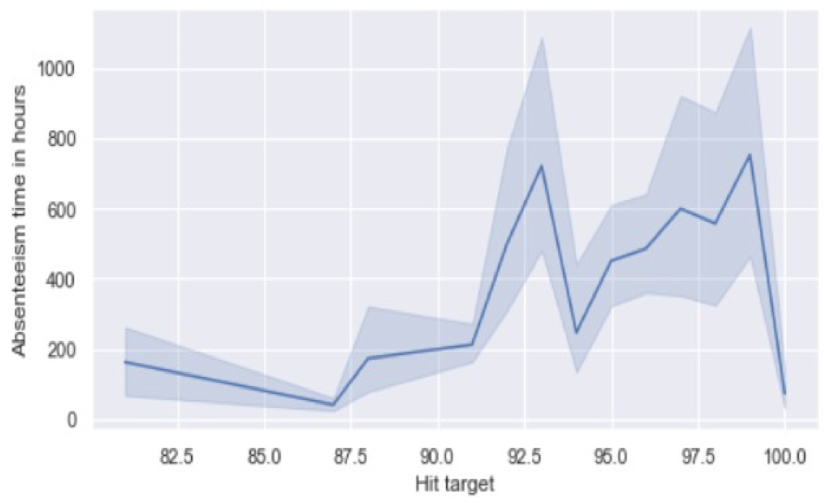
3. Increased hit target, Increased absenteeism

Employees who have hit target most has increased absenteeism. Company should assign target in such a way that

"Employees achieving high target should be assigned more target with extra compensation"

Take a look at the below graph:

```
sns.lineplot(x = 'Hit target', y = 'Absenteeism time in hours', data = Data,estimator=sum)
```



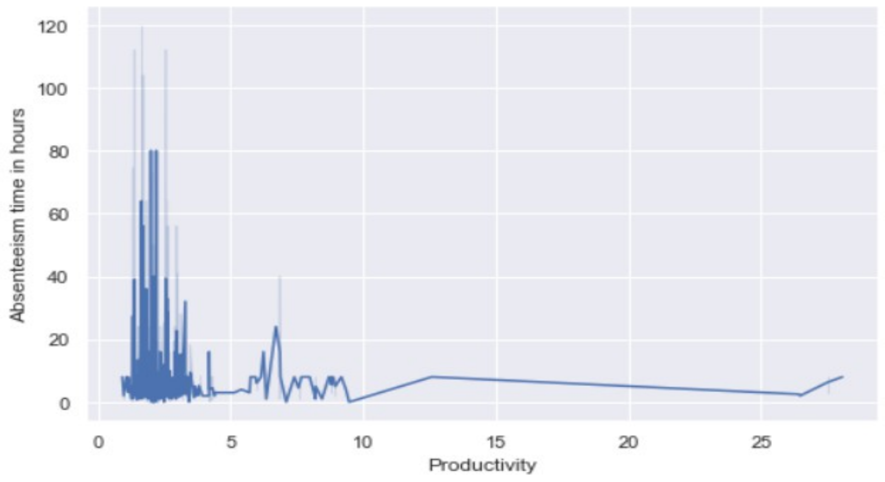
Since they are achieving their target, they are having leave easily. Instead if we assign extra target with extra compensation, they will work more productively and can eliminate thought of absence.
In addition, Let's define the new column productivity. Productivity will be calculated as below:

```
Data['Productivity'] = (Data['Work load Average/day '])/(Data['Service time']*10000)
sns.lineplot(x = 'Productivity', y = 'Absenteeism time in hours', data = Data)
```

Productivity = Workload/Service time

10000 - just a random scalar value which scales down the range of productivity values.

Take a look below for the productivity graph:



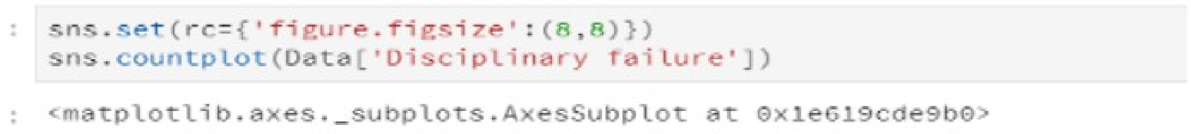
This reveals that employee with high productivity will more tend to absence.

As already said,

"Low productivity people should be recognized by company and re-skill them o enhance overall profit of the company"

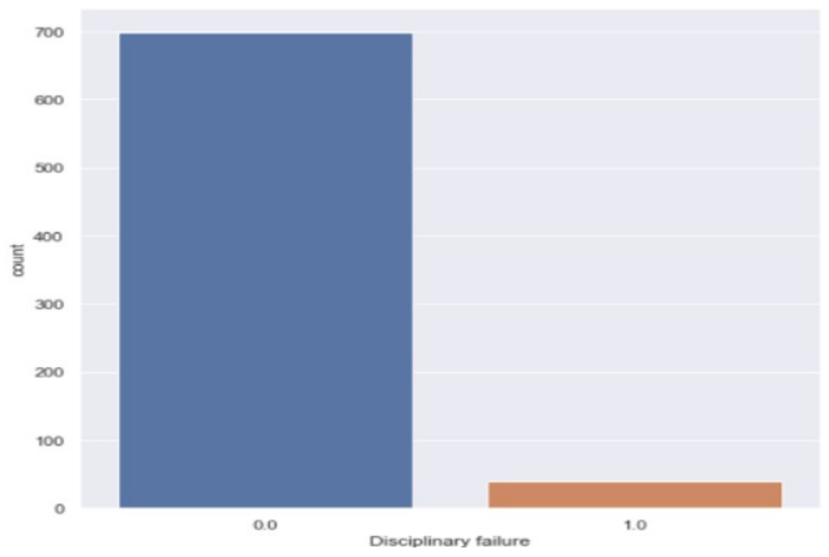
This will make them feel attached to the company and encourage the whole workforce to be more productive.

4. **Less absence of employees who underwent disciplinary action.**



```
sns.set(rc={'figure.figsize':(8,8)})
sns.countplot(Data['Disciplinary failure'])
```

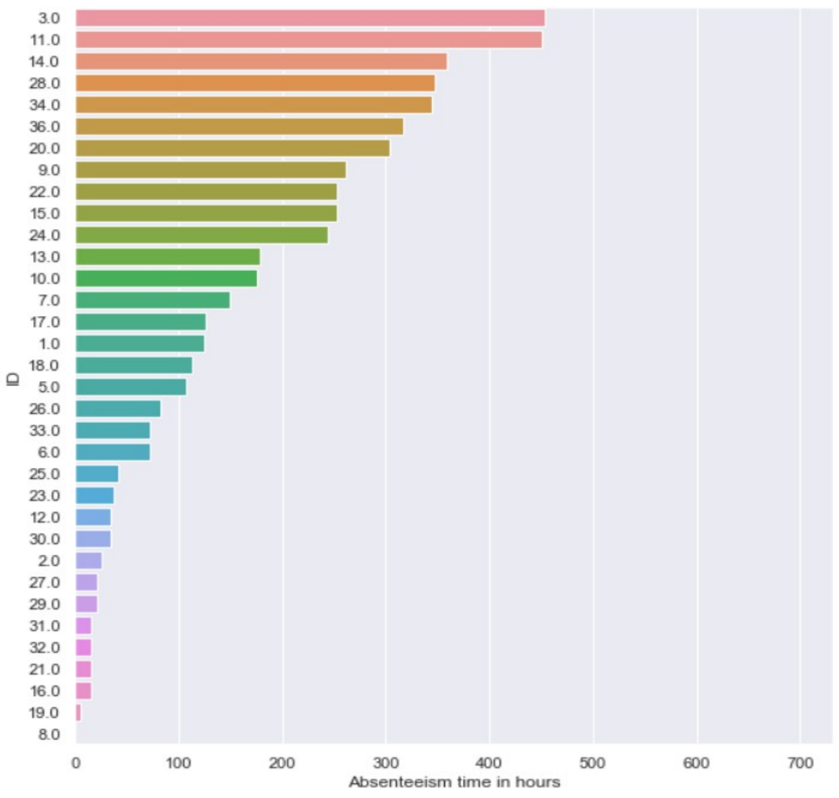
```
<matplotlib.axes._subplots.AxesSubplot at 0x1e619cde9b0>
```



Employees who has affected by Disciplinary action previously has low absence.

“Company should keenly watch employees and take disciplinary action upon employees with higher absenteeism to reduce absenteeism”

The below graph shows employee with high absent hours. Company can investigate and take action for reducing absent hours of those employees and that action wave spread through other employees too which reduces overall absenteeism.

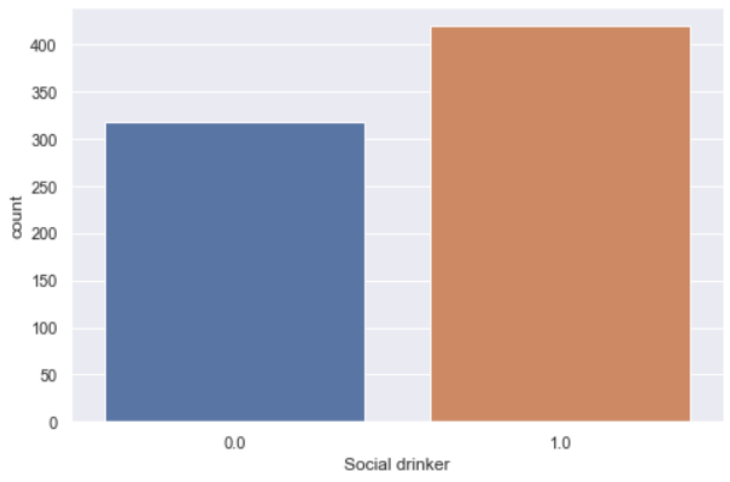


5. High absenteeism with social drinkers

Social drinkers exhibit high absenteeism. Both their work and health may affect by this drinking practice.

```
sns.countplot(Data['Social drinker'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1e61974a4a8>
```



"Company can spread general awareness to avoid drinking which works well for both company and individual"

All the above policies can be implemented in the company to lower the absenteeism rate.

Now let us continue with Outlier analysis of data. After building model, we will answer second question in problem statement.

3. Outlier analysis

Outlier values are abnormal values that are different from normal distribution. It causes data skewed in some direction. This values should be replaced with some other values within the bound.

Outlier analysis involves following steps:

1. Detecting Outliers
2. Dealing Outliers

Detecting Outliers:

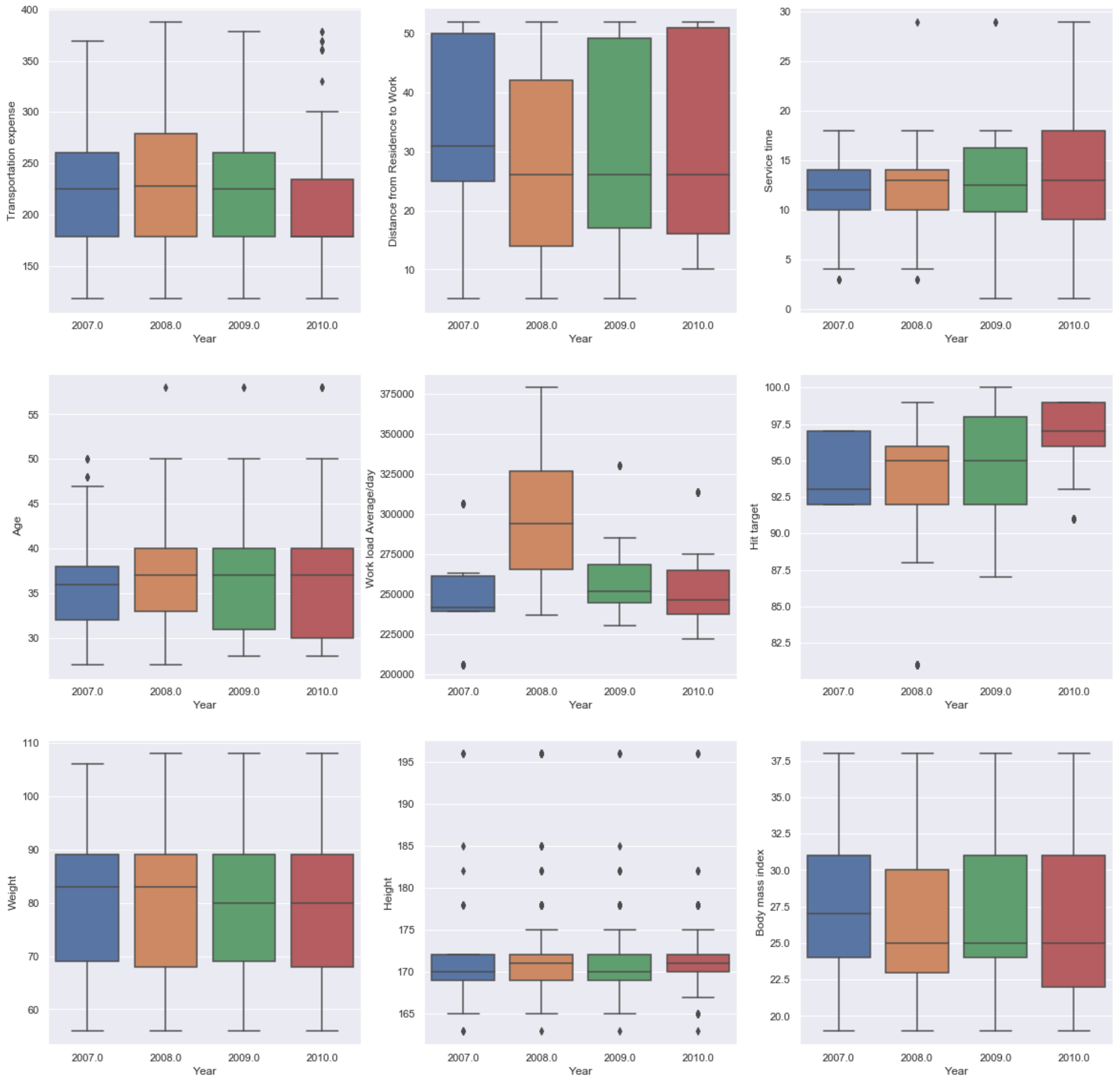
Outliers can be detected in many ways. Here we are going to use box plot method. Box plot method declares outliers as the values below 25th percentile and values above the 75th percentile including some distance (iqr*1.5 where iqr is 75th -25th).

Outlier will be detected by below method:

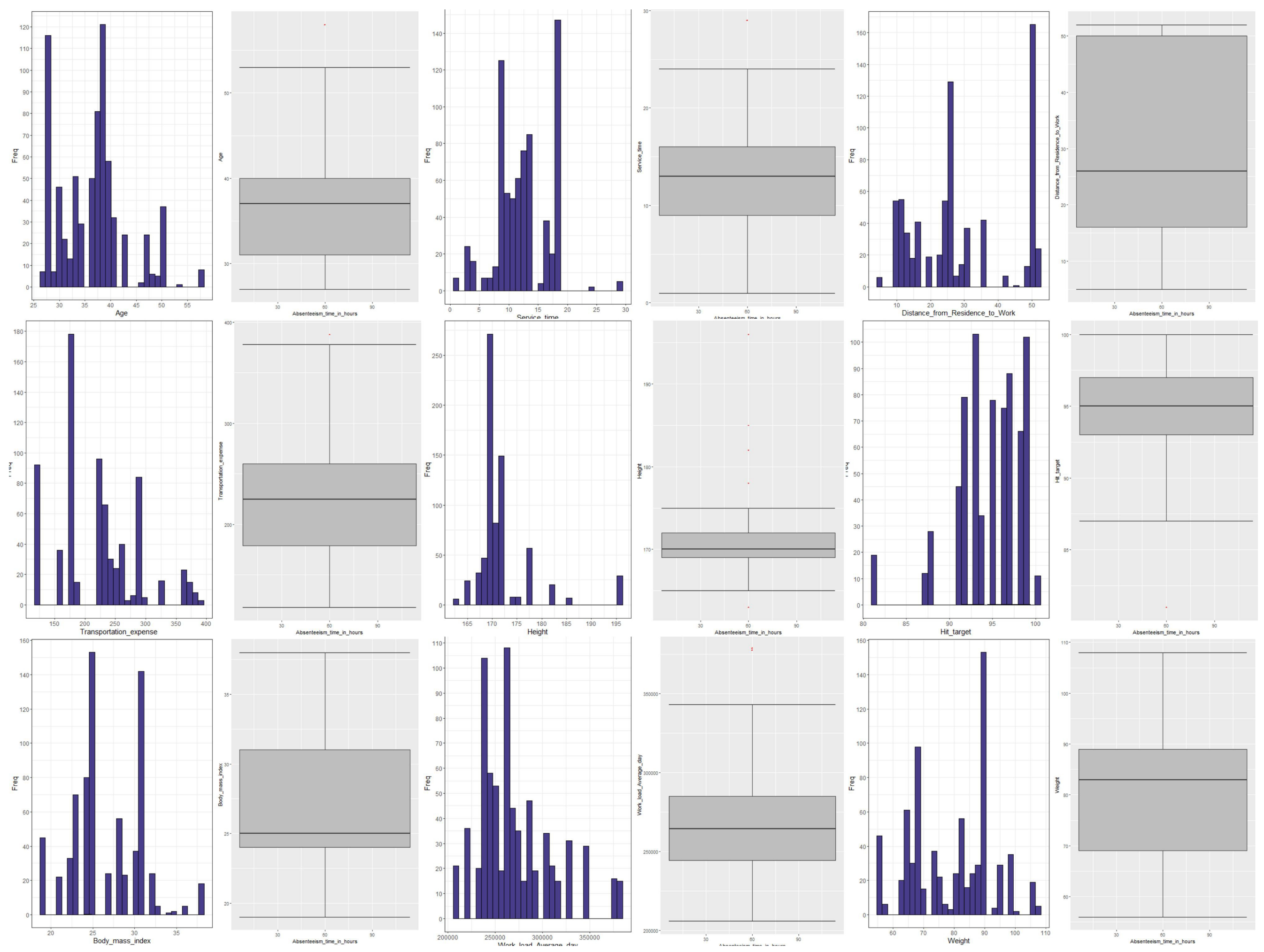
1. Box plot method – Gives column wise outliers
2. Cooks distance – Observation-wise outliers
3. OutlierTest - Observation-wise outliers

1. Box plot method:

Year-wise outlier:



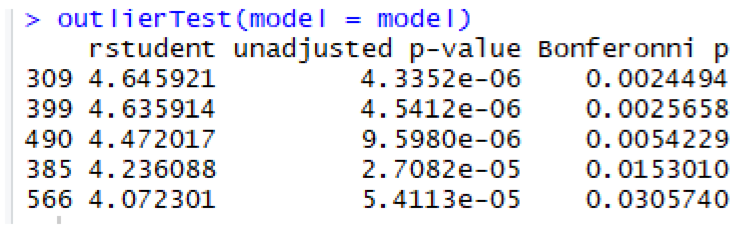
Collective Single Outlier:



2. Cooks distance and outlierTest:

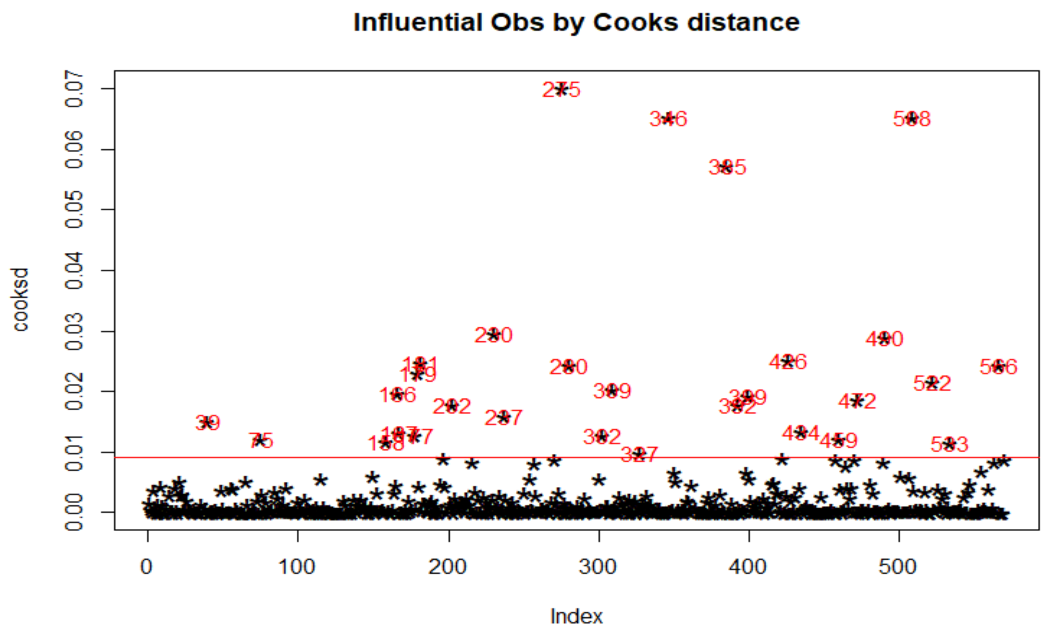
Both cooks distance and outlierTest package returns most influential or outlier ed values in the entire observation.

OutlierTest package returning outliers in train dataset.



```
> outlierTest(model = model)
    rstudent unadjusted p-value Bonferonni p
309 4.645921         4.3352e-06    0.0024494
399 4.635914         4.5412e-06    0.0025658
490 4.472017         9.5980e-06    0.0054229
385 4.236088         2.7082e-05    0.0153010
566 4.072301         5.4113e-05    0.0305740
```

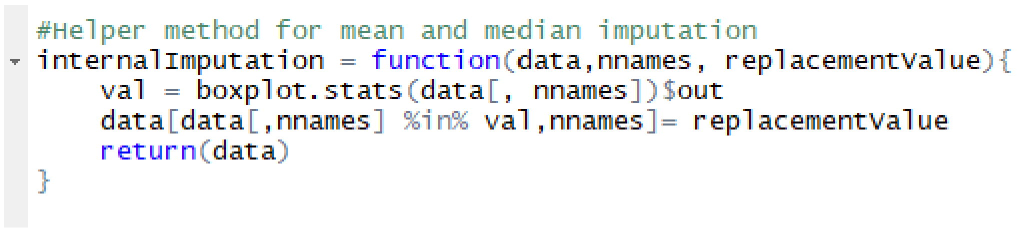
Cooks distance for most influential observations:



Dealing Outliers:

Column-wise outliers:

To deal with outliers, first of all replace all the outlier values with NA value. Take a look at the below code snippet.


```
#Helper method for mean and median imputation
internalImputation = function(data,nnames, replacementValue){
    val = boxplot.stats(data[, nnames])$out
    data[data[,nnames] %in% val,nnames]= replacementValue
    return(data)
}
```

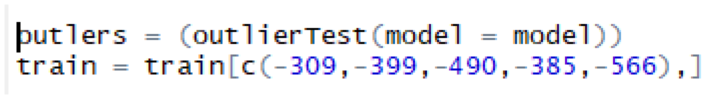
Now NA values can be replaced with below meaningful inbound values:

1. Mean
2. Median
3. Distance based method

After checking all the above methods and their corresponding accuracy, median yields most promising value compared to other for this dataset. So replace all the NA value with median value.

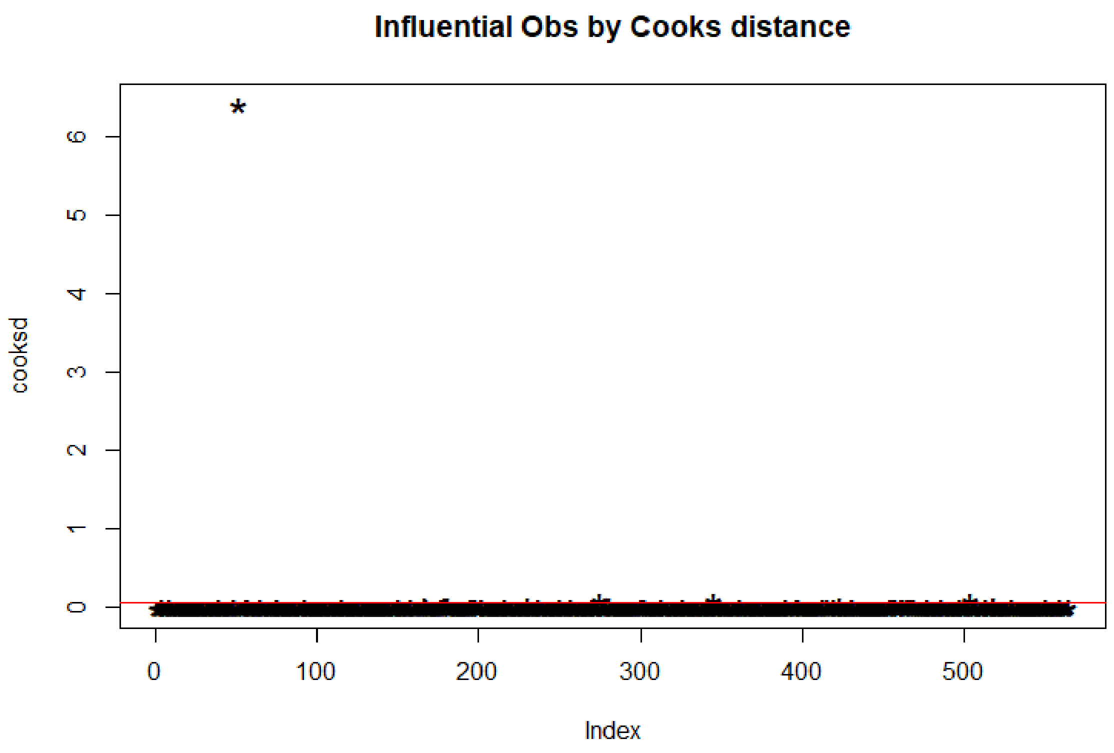
Observation-wise outliers:

Outlier generated using outlierTest package should be dropped from the training data for better model fit.



```
outlers = (outlierTest(model = model))
train = train[c(-309,-399,-490,-385,-566),]
```

After dropping these observations, if we plot cooks distance, everything will be below red-line as shown in below:



Now outliers has been removed by various means and next we can proceed to Feature selection.

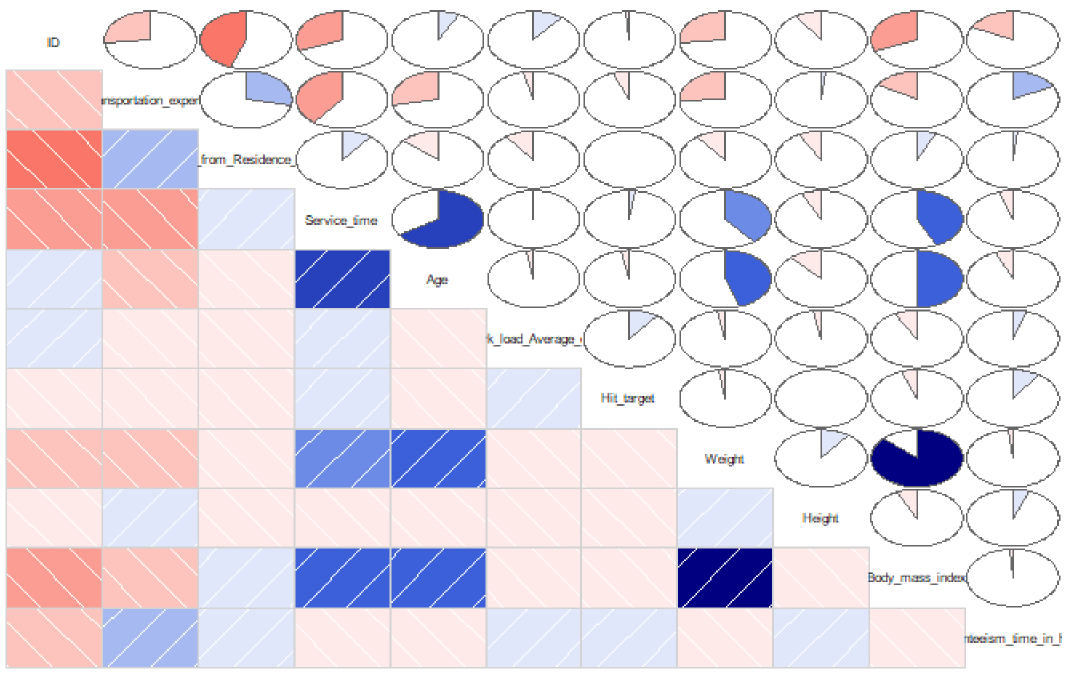
4. Feature selection

Feature selection is a technique used for selecting most important or most contributing features towards prediction. Selecting appropriate features for modeling improves model accuracy and efficiency.

Correlation analysis is an important step in feature selection. Correlation analysis helps us to detect relationship between independent predictor variables. Correlation between predictor variables will be resulting to bad accuracy because it just repeating the same information. It may also leads to over-fitting.

For exploring correlation between numerical predictors, take a look at the below plot:

Coorelation plot



We can infer the correlated values by looking the above plot.

Service time ~ Age
Weight ~ Body mass Index

The above two variables are highly correlated.

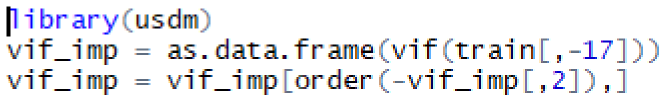
So we need to remove those variables. Among four variables, none of the variables exhibit strong correlation towards dependent variables. So, Its worth to remove all the four variables.

Then, If we notice correlation between dependent variable and all independent variables, None of the numerical variables has been strongly correlated with dependent variable.

Using Variation Inflation factor:

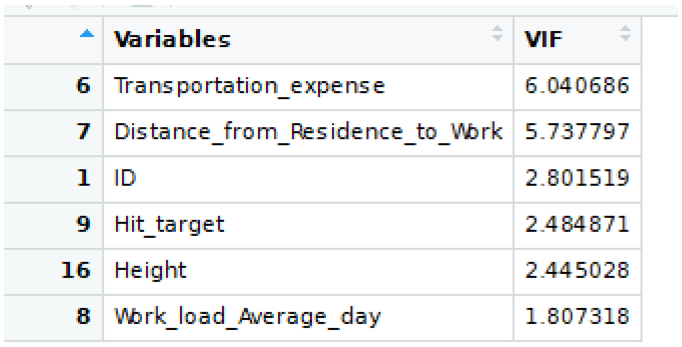
As the name suggests, a variance inflation factor (*VIF*) quantifies how much the variance is inflated. It should be in range less than 10. If any variables exhibits vif more than 10, then it should be removed.

Vif has been calculated by below:



```
library(usdm)
vif_imp = as.data.frame(vif(train[,-17]))
vif_imp = vif_imp[order(-vif_imp[,2]),]
```

Vif is mainly calculated for numerical variables. Look at the result below:

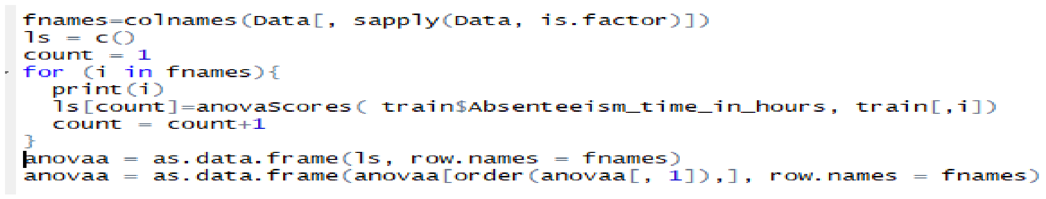


	Variables	VIF
6	Transportation_expense	6.040686
7	Distance_from_Residence_to_Work	5.737797
1	ID	2.801519
9	Hit_target	2.484871
16	Height	2.445028
8	Work_load_Average_day	1.807318

None of the variables showing values greater than 10. Show no variables will be removed using vif.

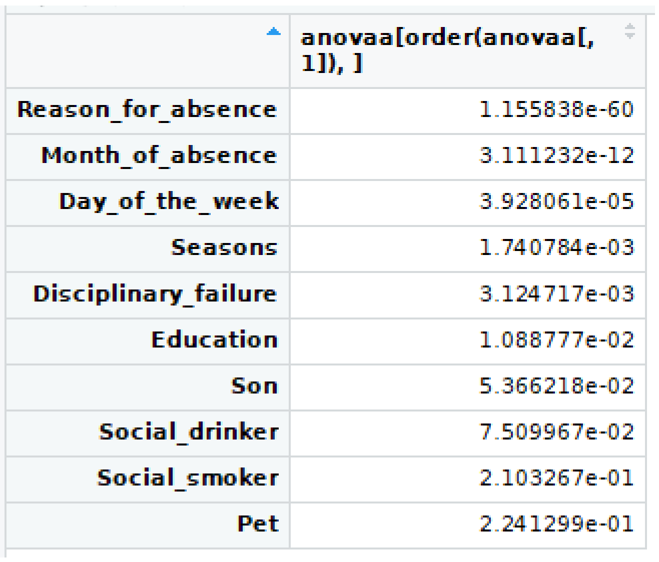
Anova test:

Anova can be used for feature selection which explains relationship between categorical variable and continuous variable. Here anovaScores() function has been used for fetch important categorical variables. If p<0.05 , Then that variable will be accepted for the further processing which is Highly correlated with dependent variable.



```
fnames=colnames(Data[, sapply(Data, is.factor)])
ls = c()
count = 1
for (i in fnames){
  print(i)
  ls[count]=anovaScores( train$Absenteeism_time_in_hours, train[,i])
  count = count+1
}
anovaa = as.data.frame(ls, row.names = fnames)
anovaa = as.data.frame(anovaa[order(anovaa[, 1]),], row.names = fnames)
```

Look at the below table for the result.



	anovaa[order(anovaa[, 1]),]
Reason_for_absence	1.155838e-60
Month_of_absence	3.111232e-12
Day_of_the_week	3.928061e-05
Seasons	1.740784e-03
Disciplinary_failure	3.124717e-03
Education	1.088777e-02
Son	5.366218e-02
Social_drinker	7.509967e-02
Social_smoker	2.103267e-01
Pet	2.241299e-01

Here (“Son”, “Social_drinker”, “Social_smoker”, “Pet”) are the variables with F-statistic or P-value greater than 0.05. Thus above are not important in predicting dependent variable and will be removed from both training and testing data.

Principal component analysis:

Principal component analysis allows us to summarize and to visualize the information in a data set containing individuals/observations described by multiple inter-correlated quantitative variables. Each variable could be considered as a different dimension.

Here, Principal components for selected features has been decreased equally from pc1 to pcn. Take a look at the below graph:

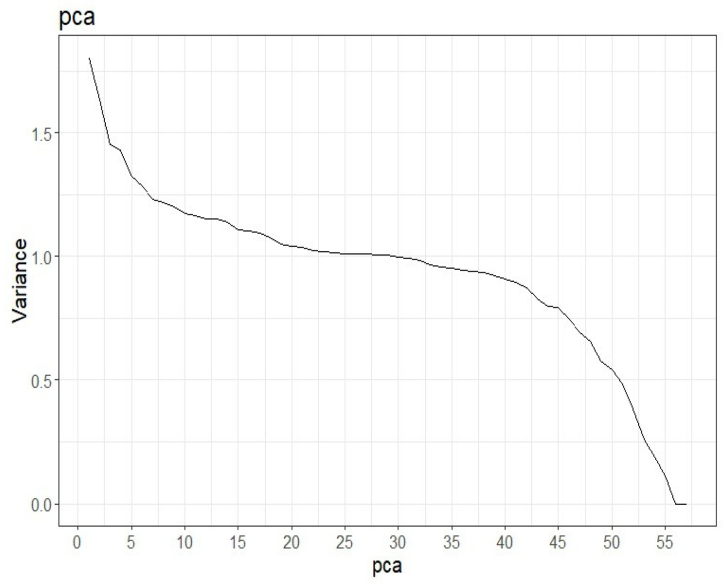


Illustration 1: With dummies

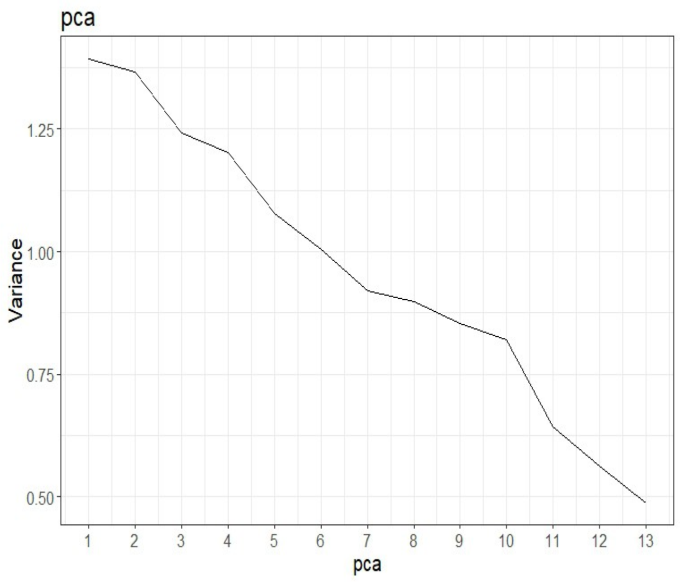


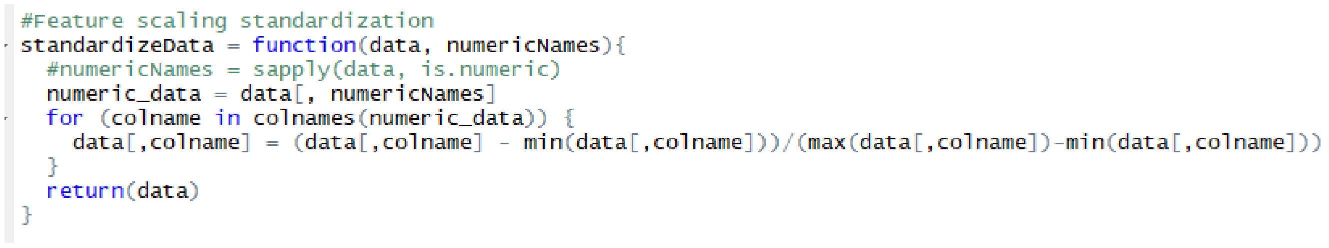
Illustration 2: Without Dummies

Dummies for categorical variables has been created and ran PCA and plot has been plotted as with dummies. By without creating, PCA has been computed and plotted as without Dummies.

Both diagram, illustrate as that there is no sharp decrease and variance has been widely explained by all components.

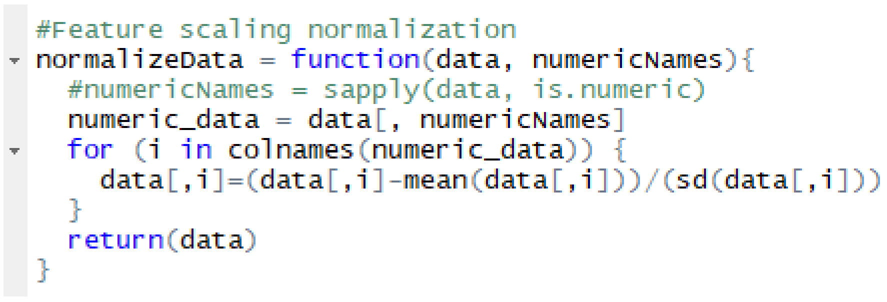
5. Feature scaling
   Feature scaling enables model to converge quickly. It is a step of Data processing which is applied to independent variables or features of data. It basically helps to normalize the data within a particular range. It also helps in speeding up the calculations in an algorithm.
   There are two ways of feature scaling:
   1. Standardization
   2. Normalization

Standardization – Standardization deals with min and range variables.



```r
#Feature scaling standardization
standardizeData = function(data, numericNames){
  #numericNames = sapply(data, is.numeric)
  numeric_data = data[, numericNames]
  for (colname in colnames(numeric_data)) {
    data[,colname] = (data[,colname] - min(data[,colname]))/(max(data[,colname])-min(data[,colname]))
  }
  return(data)
}
```

Normalization – Normalization deals with standard deviation and mean. Normalization works better for normally distributed data.



```r
#Feature scaling normalization
normalizeData = function(data, numericNames){
  #numericNames = sapply(data, is.numeric)
  numeric_data = data[, numericNames]
  for (i in colnames(numeric_data)) {
    data[,i]=(data[,i]-mean(data[,i]))/(sd(data[,i]))
  }
  return(data)
}
```

Hence, we are going to apt standardization yields good result when compared to normalization and it is not normally distributed data. Thus we are apting standardization for scaling.

Since we are going to model regression, categorical variables are replaced with its corresponding dummy variables.

Thus manipulation part of data has been completed. Now model generation will be done.

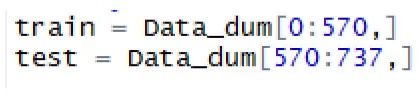
Modeling:

This problem can be dealt in both ways:

1. Regression
2. Time series analysis

For regression,

Data should be split as train and test.
To serve that purpose,
Consider, train data ranges from 2008 month 7 to 2010 month 12.
test data ranges from 2011 month 1 to 2011 month 7.



```
train = Data_dum[0:570,]
test = Data_dum[570:737,]
```

Since we considered this problem as regression problem, the algorithms that we are going to use will be below:

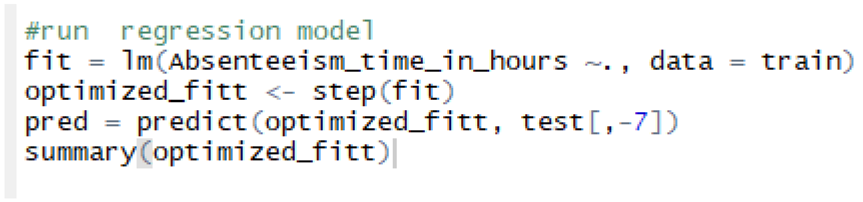
1. Linear regression
2. Linear regression with principal component analysis
3. Random Forest Regression

RMSE, MAPE measures will be used for evaluating the above models.

1. Linear regression:

Linear regression is used for finding linear relationship between target and one or more predictors.

Here, we are going to train the linear model with train data.

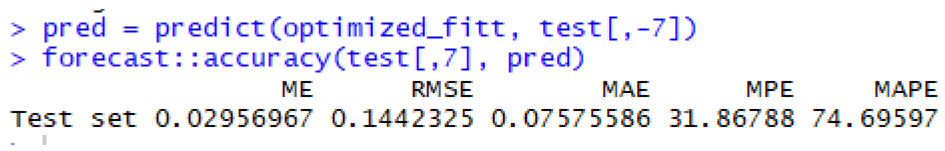


```
#run  regression model
fit = lm(Absenteeism_time_in_hours ~., data = train)
optimized_fitt <- step(fit)
pred = predict(optimized_fitt, test[,-7])
summary(optimized_fitt)
```

model will be trained with training data and intercepts and coefficients will be stored in "base" value. Here we can go for best fit model using step function. It takes base model and iteratively builds other models with all possible combinations and outputs best model and stores in optimized_fitt variable.

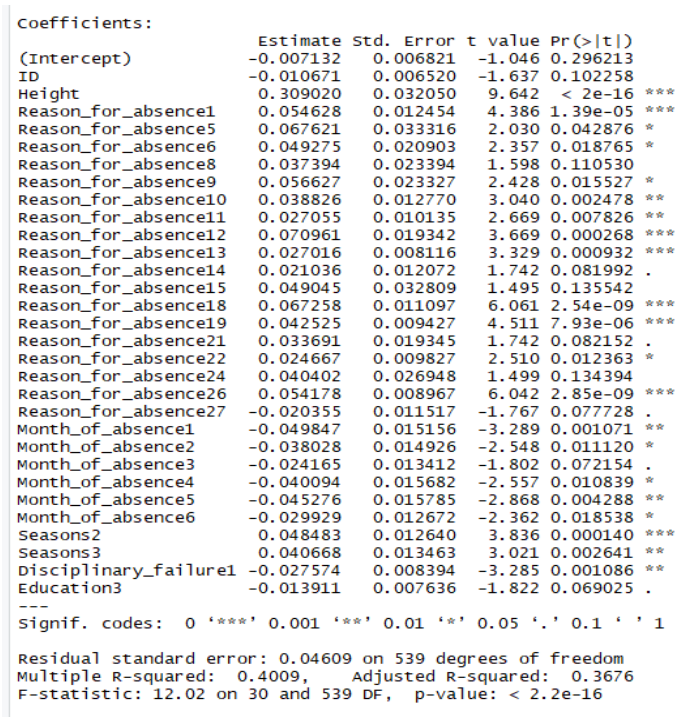
Using optimized model, We can predict absenteeism in test data.

Prediction summary as below:



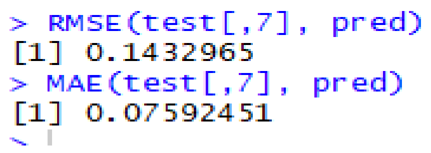
```
> pred = predict(optimized_fitt, test[,-7])
> forecast::accuracy(test[,7], pred)
                 ME      RMSE        MAE      MPE     MAPE
Test set 0.02956967 0.1442325 0.07575586 31.86788 74.69597
```

Optimized linear regression uses below variables in linear model:



```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            -0.007132   0.006821  -1.046 0.296213    
ID                     -0.010671   0.006520  -1.637 0.102258    
Height                  0.309020   0.032050   9.642  < 2e-16 ***
Reason_for_absence1     0.054628   0.012454   4.386 1.39e-05 ***
Reason_for_absence5     0.067621   0.033316   2.030 0.042876 *  
Reason_for_absence6     0.049275   0.020903   2.357 0.018765 *  
Reason_for_absence8     0.037394   0.023394   1.598 0.110530    
Reason_for_absence9     0.056627   0.023327   2.428 0.015527 *  
Reason_for_absence10    0.038826   0.012770   3.040 0.002478 ** 
Reason_for_absence11    0.027055   0.010135   2.669 0.007826 ** 
Reason_for_absence12    0.070961   0.019342   3.669 0.000268 ***
Reason_for_absence13    0.027016   0.008116   3.329 0.000932 ***
Reason_for_absence14    0.021036   0.012072   1.742 0.081992 .  
Reason_for_absence15    0.049045   0.032809   1.495 0.135542    
Reason_for_absence18    0.067258   0.011097   6.061 2.54e-09 ***
Reason_for_absence19    0.042525   0.009427   4.511 7.93e-06 ***
Reason_for_absence21    0.033691   0.019345   1.742 0.082152 .  
Reason_for_absence22    0.024667   0.009827   2.510 0.012363 *  
Reason_for_absence24    0.040402   0.026948   1.499 0.134394    
Reason_for_absence26    0.054178   0.008967   6.042 2.85e-09 ***
Reason_for_absence27   -0.020355   0.011517  -1.767 0.077728 .  
Month_of_absence1      -0.049847   0.015156  -3.289 0.001071 ** 
Month_of_absence2      -0.038028   0.014926  -2.548 0.011120 *  
Month_of_absence3      -0.024165   0.013412  -1.802 0.072154 .  
Month_of_absence4      -0.040094   0.015682  -2.557 0.010839 *  
Month_of_absence5      -0.045276   0.015785  -2.868 0.004288 ** 
Month_of_absence6      -0.029929   0.012672  -2.362 0.018538 *  
Seasons2                0.048483   0.012640   3.836 0.000140 ***
Seasons3                0.040668   0.013463   3.021 0.002641 ** 
Disciplinary_failure1  -0.027574   0.008394  -3.285 0.001086 ** 
Education3             -0.013911   0.007636  -1.822 0.069025 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04609 on 539 degrees of freedom
Multiple R-squared:  0.4009,	Adjusted R-squared:  0.3676 
F-statistic: 12.02 on 30 and 539 DF,  p-value: < 2.2e-16
```

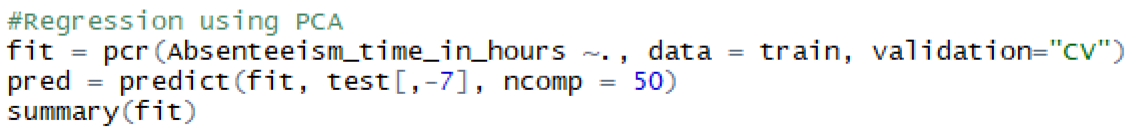
Error metrics are: RMSE, MAE



```
> RMSE(test[,7], pred)
[1] 0.1432965
> MAE(test[,7], pred)
[1] 0.07592451
```

2. Linear Regression with PCA:

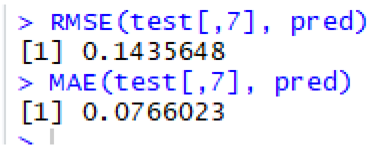
   This linear regression will be done using pca components instead of real dimensions.



```
#Regression using PCA
fit = pcr(Absenteeism_time_in_hours ~., data = train, validation="CV")
pred = predict(fit, test[,-7], ncomp = 50)
summary(fit)
```

This pcr function internally computes principal components and use that components for regression. While prediction we need to give number of components to be considered for prediction.
As we previously seen, Our PCA graph almost down after 50.
So, ncomp = 50 has been given.

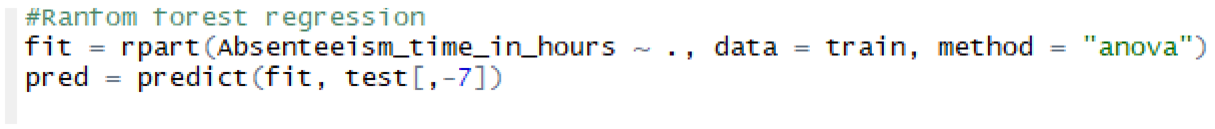
Error metrics value: RMSE and MAE



```
> RMSE(test[,7], pred)
[1] 0.1435648
> MAE(test[,7], pred)
[1] 0.0766023
```

3. Random forest regression:

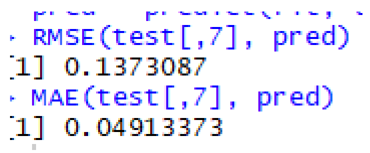
   Random forest regression uses random forest technique for predicting future values.



```
#Ranfom forest regression
fit = rpart(Absenteeism_time_in_hours ~ ., data = train, method = "anova")
pred = predict(fit, test[,-7])
```

Same test and training data has been used through out all the model.

Error metric: RMSE and MAE



```
> RMSE(test[,7], pred)
[1] 0.1373087
> MAE(test[,7], pred)
[1] 0.04913373
```

Let's compare the error metrics in tabular form:

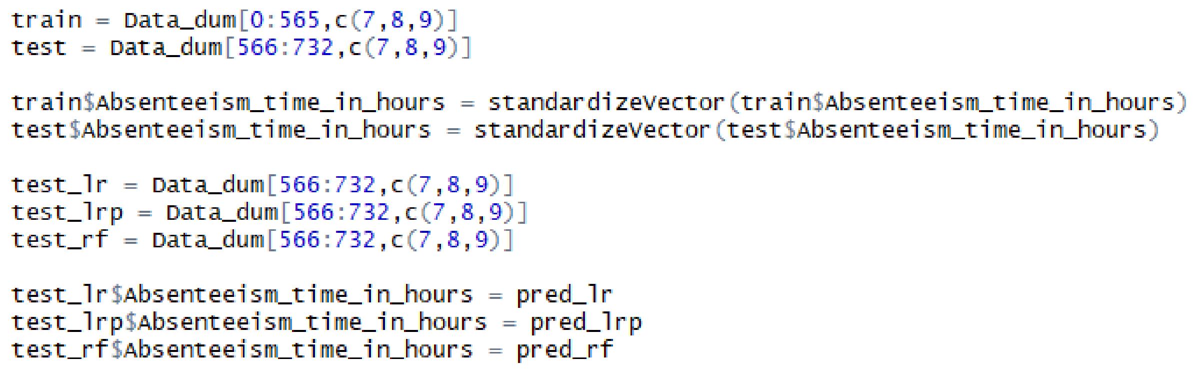
Algorithms	RMSE	MAE
Linear regression	0.1432965	0.07592451
Linear regression with pca	0.1435648	0.0766023
Random forest regression	0.1373087	0.04913373

Random Forest regression seams best among all the algorithms. We can decide which is best among three by plotting the forcasts with actual data. But since it is time-series data, we have to try Arimax modelling too.

Forecasting for 2011:

Let's forecast output with predicted data.

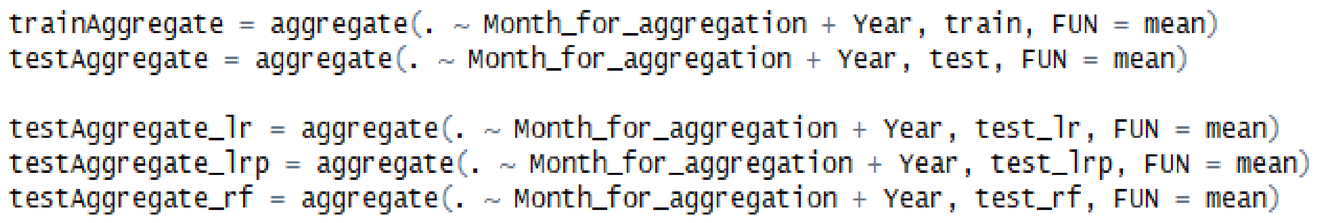
Let's build train and test data with Absenteeism_time_in_hours, Year, Month_of_absence.



```
train = Data_dum[0:565,c(7,8,9)]
test = Data_dum[566:732,c(7,8,9)]

train$Absenteeism_time_in_hours = standardizeVector(train$Absenteeism_time_in_hours)
test$Absenteeism_time_in_hours = standardizeVector(test$Absenteeism_time_in_hours)

test_lr = Data_dum[566:732,c(7,8,9)]
test_lrp = Data_dum[566:732,c(7,8,9)]
test_rf = Data_dum[566:732,c(7,8,9)]

test_lr$Absenteeism_time_in_hours = pred_lr
test_lrp$Absenteeism_time_in_hours = pred_lrp
test_rf$Absenteeism_time_in_hours = pred_rf
```

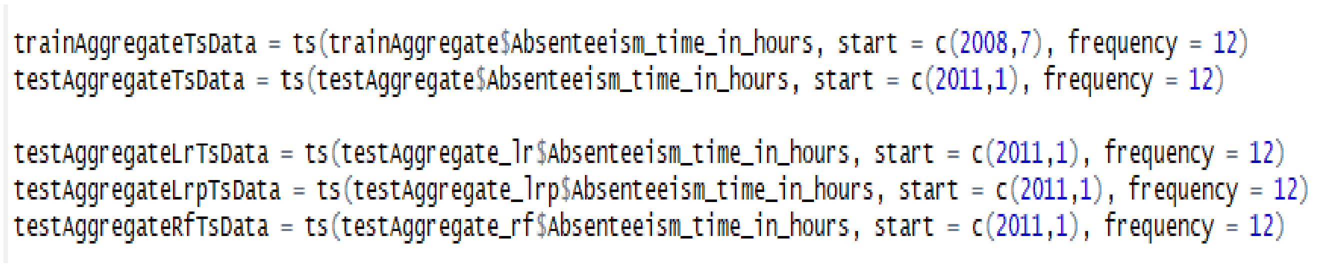
Build Linear prediction data, Linear Regression with PCA and Random forest prediction data.

Then aggregate Absenteeism_time_in_hours with Year and Month Value.



```
trainAggregate = aggregate(. ~ Month_for_aggregation + Year, train, FUN = mean)
testAggregate = aggregate(. ~ Month_for_aggregation + Year, test, FUN = mean)

testAggregate_lr = aggregate(. ~ Month_for_aggregation + Year, test_lr, FUN = mean)
testAggregate_lrp = aggregate(. ~ Month_for_aggregation + Year, test_lrp, FUN = mean)
testAggregate_rf = aggregate(. ~ Month_for_aggregation + Year, test_rf, FUN = mean)
```

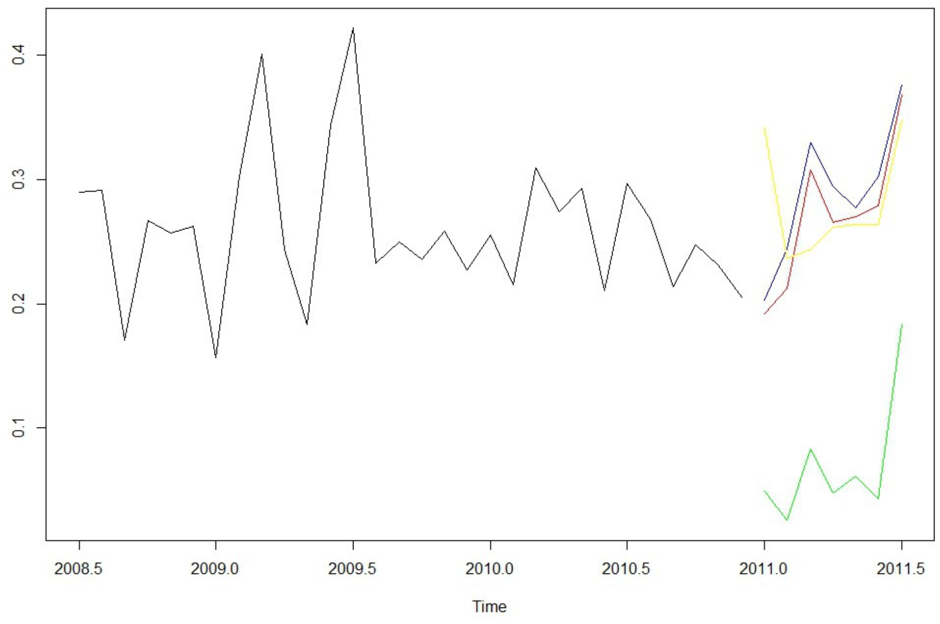
Change dataframe into timeseries by specifying starting year and month.



```
trainAggregateTsData = ts(trainAggregate$Absenteeism_time_in_hours, start = c(2008,7), frequency = 12)
testAggregateTsData = ts(testAggregate$Absenteeism_time_in_hours, start = c(2011,1), frequency = 12)

testAggregateLrTsData = ts(testAggregate_lr$Absenteeism_time_in_hours, start = c(2011,1), frequency = 12)
testAggregateLrpTsData = ts(testAggregate_lrp$Absenteeism_time_in_hours, start = c(2011,1), frequency = 12)
testAggregateRfTsData = ts(testAggregate_rf$Absenteeism_time_in_hours, start = c(2011,1), frequency = 12)
```

Plot all the build data.

```
ts.plot(trainAggregateTsData, testAggregateTsData, testAggregateLrTsData,testAggregateLrpTsData,
        testAggregateRfTsData,gpars = list(col = c("black","green", "red", "blue", "yellow")))
```



The above graph forecasts 2011 values. Since it is a regression, It just follows the same pattern as original value.

Green color line is the actual value. Red, blue and yellow are predicted value 2011 of linear regression, linear regression with pca and Random forest models.

ARIMA – Auto Regressive Integrated Moving Average

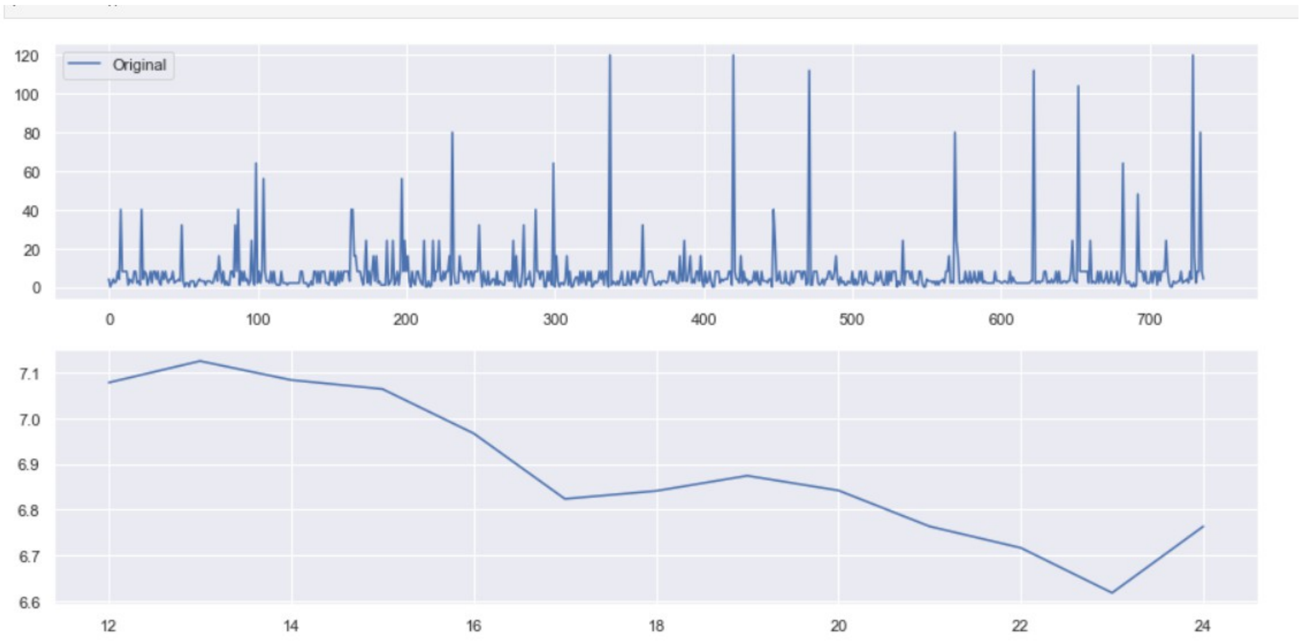
Arima forecasting has been used for uni variate and multivariate analysis of time-series. Arima with xreg field is used for multivariate analysis.

```
arim = auto.arima(trainAggregateTsData, trace=TRUE)
fcast_cons <- forecast(arim, h = 7)
autoplot(fcast_cons)
```

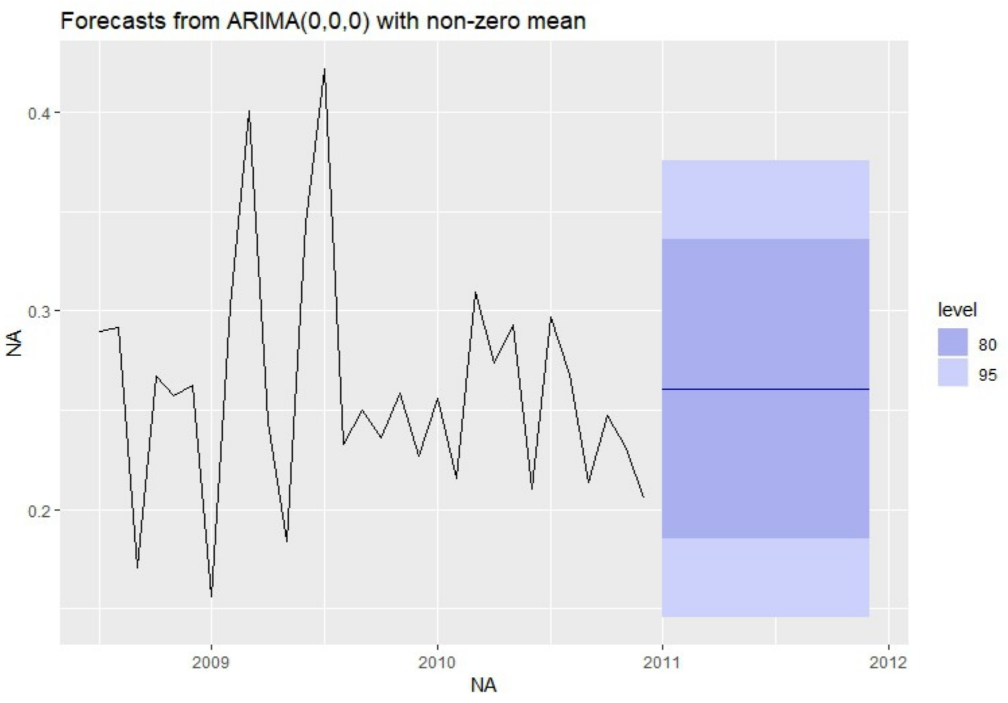
auto-arima chooses automatically p, q and d. Here ideal p, q, and d values are 0,0,0.

```
> forecast::accuracy(test$Absenteeism_time_in_hours, fcast_cons$mean)
                ME      RMSE       MAE      MPE     MAPE      ACF1 Theil's U
Test set 0.1926523 0.2039793 0.1926523 73.95214 73.95214 0.3933303       Inf
```

By univariate analysis, Employee absenteeism has decreasing trend. Hence in 2011 absenteeism decreases further.



Forcasting of future values of univariate analysis was just around the mean usually.


Forecasts from ARIMA(0,0,0) with non-zero mean
NA
NA
level