



ACADGILD

Assignment

SESSION: 16 To 20
Jagannath Chaudhary –
B4

Assignment

Index:

Task 16 (1)– Page – 3 to 76

Task 17(2) -Page – 77 to 100

Task 18 (3)-Page- 101 to 120

Task 19(4) – Page – 121 to 156

Task 20(5) – Page – 157 to 179

Table of Contents

1. Introduction	2
2. Objective	2
3. Prerequisites	2
4. Associated Data Files	2
5. Problem Statement	3
6. Expected Output	169

. Introduction

This assignment will help you understand the concepts learned in the session.

1. **Objective**

This assignment will test your skills on the concepts of SQL analytics.

1. **Prerequisites**

Not applicable.

1. **Associated Data Files**

Not applicable.

1. Problem Statement

Task 1:

1. Use the below-given data set

Data Set

2. Perform the below-given activities:
 - a. Predict the no of comments in next H hrs

Note:-

1. Use LASSO, Elastic Net and Ridge and other regression techniques that are covered in the module
2. Report the training accuracy and test accuracy
3. compare with linear models and report the accuracy
4. create a graph displaying the accuracy of all models

```
library(readr)
library(data.table)
library(tidyr)
install.packages("tidyverse")
library(tidyverse)
library(caret)
library(glmnet)
install.packages("mlbench")
library(mlbench)
getwd()
```

```

data1<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_1.csv")
data2<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_2.csv")
data3<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_3.csv")
data4<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_4.csv")
data5<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_5.csv")
train_set<-rbind(data1,data2,data3,data4,data5)
train_set1<-na.omit(train_set)
train_set1
#data(train_set1, package="mlbench")

```

```

> train_set1

```

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13					
1:	634995	0	463	1	0	806	11.29104	1	70.49514	0	806	7.574627	0	69.				
2:	634995	0	463	1	0	806	11.29104	1	70.49514	0	806	7.574627	0	69.				
3:	634995	0	463	1	0	806	11.29104	1	70.49514	0	806	7.574627	0	69.				
4:	634995	0	463	1	0	806	11.29104	1	70.49514	0	806	7.574627	0	69.				
5:	634995	0	463	1	0	806	11.29104	1	70.49514	0	806	7.574627	0	69.				

602809:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0	1990	229.165289	79	386.				
602810:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0	1990	229.165289	79	386.				
602811:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0	1990	229.165289	79	386.				
602812:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0	1990	229.165289	79	386.				
602813:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0	1990	229.165289	79	386.				
	V16	V17	V18			V19	V20	V21	V22	V23	V24	V25	V26					
1:	76	2.604478	0			8.505502	0	806	10.64925	1	70.25479	-69	806	4.970				
2:	76	2.604478	0			8.505502	0	806	10.64925	1	70.25479	-69	806	4.970				
3:	76	2.604478	0			8.505502	0	806	10.64925	1	70.25479	-69	806	4.970				
4:	76	2.604478	0			8.505502	0	806	10.64925	1	70.25479	-69	806	4.970				
5:	76	2.604478	0			8.505502	0	806	10.64925	1	70.25479	-69	806	4.970				

602809:	1734	174.595041	60	298.763750		0	2082	446.80165	240	472.06568	-1348	1990	54.570					
602810:	1734	174.595041	60	298.763750		0	2082	446.80165	240	472.06568	-1348	1990	54.570					
602811:	1734	174.595041	60	298.763750		0	2082	446.80165	240	472.06568	-1348	1990	54.570					
602812:	1734	174.595041	60	298.763750		0	2082	446.80165	240	472.06568	-1348	1990	54.570					
602813:	1734	174.595041	60	298.763750		0	2082	446.80165	240	472.06568	-1348	1990	54.570					
	V29	V30	V31	V32	V33	V34	V35	V36	V37	V38	V39	V40	V41	V42	V43	V44	V45	V46
1:	69.85058	0	0	0	0	0	65	166	2	0	24	0	0	0	1	0	0	0
2:	69.85058	0	0	0	0	0	10	132	1	0	24	0	0	0	0	1	0	0
3:	69.85058	0	0	0	0	0	14	133	2	0	24	0	0	0	0	0	1	0
4:	69.85058	7	0	3	7	-3	62	131	1	0	24	0	0	0	0	0	1	0
5:	69.85058	1	0	0	1	0	58	142	5	0	24	0	1	0	0	0	0	0

602809:	506.79360	1940	1940	0	1940	1940	16	77	2123	0	24	0	0	1	0	0	0	0
602810:	506.79360	155	155	0	155	155	4	130	474	0	24	0	0	1	0	0	0	0
602811:	506.79360	207	207	0	207	207	12	12	1511	0	24	0	0	0	1	0	0	0
602812:	506.79360	629	85	544	619	-459	31	149	1099	0	24	0	0	0	1	0	0	0

#2. Perform the below given activities:

a. Predict the no of comments in next H hrs

```
x<-as.matrix(fbmsg[,1:53])
```

```
y<-as.matrix(fbmsg[,54])
```

#####Note:-

#1. Use LASSO, Elastic Net and Ridge and other regression techniques that are covered in the module

```
library(glmnet)
```

```
fit_ridge<-glmnet(x,y, family = "gaussian", alpha = 0, lambda = 0.001)
```

```
lasso
```

```
summary(fit_ridge)
```

```
> summary(fit_ridge)
      Length Class      Mode
a0          1  -none-    numeric
beta        53 dgCMatrix S4
df           1  -none-    numeric
dim          2  -none-    numeric
lambda       1  -none-    numeric
dev.ratio    1  -none-    numeric
nulldev      1  -none-    numeric
npasses      1  -none-    numeric
jerr         1  -none-    numeric
offset       1  -none-    logical
call         6  -none-    call
nobs         1  -none-    numeric
```

```
predictions1<-predict(lasso, x, type = "link")
```

```
mse1<-mean((y-predictions1)^2)
```

```
mse1
```

```
library(glmnet)
```

```
fit_lasso<-glmnet(x,y, family = "gaussian", alpha = 1, lambda = 0.001)
```

```
fit_lasso
```

```
summary(fit_lasso)
```

```
> fit_lasso
Call:  glmnet(x = x, y = y, family = "gaussian", alpha = 1, lambda = 0.001)
```

```

      Df    %Dev Lambda
[1,] 46 0.5534 0.001
      Length Class      Mode
a0         1    -none-  numeric
beta       53    dgCMatrix S4
df          1    -none-  numeric
dim         2    -none-  numeric
lambda      1    -none-  numeric
dev.ratio   1    -none-  numeric
nulldev     1    -none-  numeric
npasses     1    -none-  numeric
jerr        1    -none-  numeric
offset      1    -none-  logical
call        6    -none-   call
nobs        1    -none-  numeric

```

```
predictions2<-predict(fit_lasso, x, type = "link")
```

```
mse2<-mean((y-predictions2)^2)
```

```
mse2
```

```

predictions2<-predict(fit_lasso, x, type = "link")
> mse2<-mean((y-predictions2)^2)
> mse2
[1] 3403.824

```

```
fit_elnet<-glmnet(x,y, family = "gaussian", alpha = 0.5, lambda = 0.001)
```

```
fit_elnet
```

```
summary(fit_elnet)
```

```
predictions3<-predict(fit_elnet, x, type = "link")
```

```
mse3<-mean((y-predictions3)^2)
```

```
mse3
```

```

> fit_elnet<-glmnet(x,y, family = "gaussian", alpha = 0.5, lambda = 0.001)
> fit_elnet

Call:  glmnet(x = x, y = y, family = "gaussian", alpha = 0.5, lambda = 0.001)

      Df    %Dev Lambda
[1,] 46 0.5531 0.001
> summary(fit_elnet)
      Length Class      Mode
a0         1    -none-  numeric
beta       53    dgCMatrix S4
df          1    -none-  numeric
dim         2    -none-  numeric
lambda      1    -none-  numeric
dev.ratio   1    -none-  numeric
nulldev     1    -none-  numeric
npasses     1    -none-  numeric
jerr        1    -none-  numeric

```

```

offset      1      -none-      logical
call        6      -none-      call
nobs        1      -none-      numeric
> predictions3<-predict(fit_elfnet, x, type = "link")
> mse3<-mean((y-predictions3)^2)
> mse3
[1] 3406.419

```

10-fold Cross validation for each alpha = 0, 0.1, ... , 0.9, 1.0

```

for (i in 0:10) {
  assign(paste("fit", i, sep=""), cv.glmnet(x, y, type.measure="mse",
      alpha=i/10,family="gaussian"))
}

```

Plot solution paths:

```
par(mfrow=c(3,2))
```

```

library(earth)
model1<-earth(V54~., data = fbmsg)
model1

```

```

summary(model1)
evimp(model1)
predictions4<-predict(model1, fbmsg)

```

```

mse4<-mean((fbmsg$V54-predictions4)^2)
mse4

```

```

model1<-earth(V54~., data = fbmsg)
> model1
Selected 18 of 22 terms, and 10 of 53 predictors
Termination condition: RSq changed by less than 0.001 at 22 terms
Importance: V31, V35, V12, V13, V29, V10, V22, V8, V39, V14, V1-unused, V2-unused, V3-unused, ...
Number of terms at each degree of interaction: 1 17 (additive model)
GCV 3023.449      RSS 1608601      GRSq 0.6046489      RSq 0.6482564
> summary(model1)
Call: earth(formula=V54~., data=fbmsg)

```



```

              coefficients
(Intercept)    228.272462
h(301-V8)       0.643775
h(V8-301)      -0.317003
h(V10-16)       1.216314
h(278.343-V12)  0.819529
h(V12-278.343) -2.757007
h(V13-254.5)    2.683075
h(354.967-V14) -0.486345
h(V14-354.967) -1.255793
h(362.778-V22) -1.220286
h(V22-362.778) -0.599143
h(V29-252.129) -2.149127
h(298.581-V29)  0.544231
h(V29-298.581)  4.089788
h(618-V31)     -0.334238
h(V31-618)     0.128075
h(6-V35)       11.511868
h(5-V39)       -8.092162

Selected 18 of 22 terms, and 10 of 53 predictors
Termination condition: RSq changed by less than 0.001 at 22 terms
Importance: V31, V35, V12, V13, V29, V10, V22, V8, V39, V14, V1-unused, V2-unused, V3-unused, ...
Number of terms at each degree of interaction: 1 17 (additive model)
GCV 3023.449    RSS 1608601    GRSq 0.6046489    RSq 0.6482564
> evimp(modell1)
      nsubsets    gcv    rss
V31         17 100.0  100.0
V35         15  62.7   65.0
V12         14  55.1   57.6
V13         14  55.1   57.6
V29         14  55.1   57.6
V10         11  34.0   38.3
V22          9  29.6   33.6
V8           8  29.0   32.4
V39          7  25.7   29.1
V14          5  19.7   22.9
> predictions4<-predict(modell1, fbmsg)
> mse4<-mean((fbmsg$V54-predictions4)^2)
> mse4
[1] 2681.001

```

step wise regression

```
base<-lm(V54~., fbmsg)
```

```
base
```

```
summary(base)
```

```
fitt<-step(base)
```

```
summary(fitt)
```

```
prediction5<-predict(fitt, fbmsg)
```

```
mse5<-mean((fbmsg$V54-prediction5)^2)
```

```
mse5
```

```
> fitt<-step(base)
Start:  AIC=4952.96
V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
      V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
      V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
      V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
      V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V50 + V51 +
      V52 + V53

Step:  AIC=4952.96
V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
      V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
      V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
      V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
      V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V50 + V51 +
      V52

Step:  AIC=4952.96
V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
      V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
      V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
      V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
      V42 + V43 + V44 + V45 + V46 + V47 + V48 + V49 + V51 + V52

Step:  AIC=4952.96
V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
      V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
      V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
      V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
      V42 + V43 + V44 + V45 + V46 + V47 + V48 + V51 + V52

Step:  AIC=4952.96
V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
      V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
      V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
      V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
      V42 + V43 + V44 + V45 + V47 + V48 + V51 + V52

Step:  AIC=4952.96
V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
      V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
      V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
      V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
      V42 + V44 + V45 + V47 + V48 + V51 + V52
```

Step: AIC=4952.96

V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
V32 + V33 + V34 + V35 + V36 + V37 + V38 + V39 + V40 + V41 +
V44 + V45 + V47 + V48 + V51 + V52

Step: AIC=4952.96

V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
V32 + V33 + V34 + V35 + V36 + V37 + V39 + V40 + V41 + V44 +
V45 + V47 + V48 + V51 + V52

Step: AIC=4952.96

V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
V22 + V23 + V24 + V25 + V26 + V27 + V28 + V29 + V30 + V31 +
V32 + V33 + V35 + V36 + V37 + V39 + V40 + V41 + V44 + V45 +
V47 + V48 + V51 + V52

Step: AIC=4952.96

V54 ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 + V9 + V10 + V11 +
V12 + V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 +
V22 + V23 + V24 + V25 + V26 + V28 + V29 + V30 + V31 + V32 +
V33 + V35 + V36 + V37 + V39 + V40 + V41 + V44 + V45 + V47 +
V48 + V51 + V52

	Df	Sum of Sq	RSS	AIC
- V5	1	12	1986457	4951.0
- V37	1	33	1986479	4951.0
- V12	1	64	1986509	4951.0
- V4	1	154	1986599	4951.0
- V20	1	158	1986603	4951.0
- V48	1	161	1986606	4951.0
- V33	1	170	1986615	4951.0
- V2	1	359	1986805	4951.1
- V14	1	445	1986890	4951.1
- V30	1	955	1987400	4951.2
- V51	1	1046	1987491	4951.3
- V36	1	1663	1988108	4951.5
- V41	1	1876	1988321	4951.5
- V8	1	1935	1988380	4951.5
- V29	1	2355	1988800	4951.7
- V32	1	2368	1988813	4951.7
- V40	1	2486	1988931	4951.7
- V23	1	2826	1989271	4951.8
- V31	1	3531	1989977	4952.0
- V52	1	3844	1990289	4952.1
- V44	1	3991	1990436	4952.2
- V45	1	4167	1990612	4952.2

```

- V18 1 4521 1990966 4952.3
- V47 1 6390 1992835 4952.9
<none> 1986445 4953.0
- V39 1 10499 1996944 4954.1
- V26 1 13272 1999717 4955.0
- V3 1 13426 1999871 4955.0
- V10 1 14077 2000522 4955.2
- V11 1 25422 2011867 4958.6
- V1 1 25831 2012276 4958.7
- V13 1 29954 2016399 4959.9
- V35 1 33887 2020332 4961.1
- V28 1 34523 2020969 4961.3
- V15 1 51368 2037813 4966.3
- V22 1 52383 2038828 4966.6
- V7 1 52507 2038952 4966.6
- V25 1 64996 2051441 4970.3
- V24 1 93065 2079510 4978.4
- V9 1 97134 2083579 4979.6
- V6 1 140611 2127056 4992.0
- V21 1 143623 2130068 4992.8
- V16 1 171561 2158006 5000.7
- V17 1 173228 2159673 5001.1
- V19 1 347818 2334263 5047.8

```

Step: AIC=4950.96

```

V54 ~ V1 + V2 + V3 + V4 + V6 + V7 + V8 + V9 + V10 + V11 + V12 +
      V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 +
      V23 + V24 + V25 + V26 + V28 + V29 + V30 + V31 + V32 + V33 +
      V35 + V36 + V37 + V39 + V40 + V41 + V44 + V45 + V47 + V48 +
      V51 + V52

```

	Df	Sum of Sq	RSS	AIC
- V37	1	32	1986489	4949.0
- V12	1	65	1986522	4949.0
- V4	1	151	1986608	4949.0
- V48	1	160	1986617	4949.0
- V33	1	174	1986631	4949.0
- V2	1	363	1986820	4949.1
- V14	1	441	1986898	4949.1
- V30	1	956	1987413	4949.2
- V51	1	1050	1987507	4949.3
- V36	1	1671	1988128	4949.5
- V41	1	1884	1988342	4949.5
- V8	1	2137	1988594	4949.6
- V29	1	2348	1988806	4949.7
- V32	1	2365	1988822	4949.7
- V40	1	2511	1988968	4949.7
- V23	1	3094	1989551	4949.9
- V31	1	3655	1990112	4950.1
- V52	1	3843	1990300	4950.1
- V44	1	4008	1990465	4950.2
- V45	1	4171	1990629	4950.2
- V18	1	4531	1990988	4950.3
- V47	1	6378	1992835	4950.9
<none>			1986457	4951.0
- V39	1	10491	1996949	4952.1

```

- V26 1 13412 1999869 4953.0
- V3 1 13559 2000017 4953.0
- V10 1 14940 2001397 4953.5
- V20 1 20108 2006565 4955.0
- V11 1 25582 2012039 4956.6
- V1 1 25919 2012377 4956.7
- V13 1 30140 2016597 4958.0
- V35 1 33966 2020424 4959.1
- V28 1 34933 2021390 4959.4
- V22 1 53689 2040146 4965.0
- V7 1 54088 2040545 4965.1
- V25 1 65327 2051785 4968.4
- V15 1 83658 2070115 4973.7
- V24 1 94870 2081328 4977.0
- V9 1 99314 2085772 4978.2
- V6 1 141450 2127907 4990.2
- V21 1 144088 2130545 4991.0
- V16 1 173721 2160179 4999.3
- V17 1 176592 2163049 5000.1
- V19 1 358168 2344625 5048.4

```

Step: AIC=4948.97

```

V54 ~ V1 + V2 + V3 + V4 + V6 + V7 + V8 + V9 + V10 + V11 + V12 +
      V13 + V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 +
      V23 + V24 + V25 + V26 + V28 + V29 + V30 + V31 + V32 + V33 +
      V35 + V36 + V39 + V40 + V41 + V44 + V45 + V47 + V48 + V51 +
      V52

```

	Df	Sum of Sq	RSS	AIC
- V12	1	67	1986556	4947.0
- V4	1	147	1986636	4947.0
- V48	1	157	1986646	4947.0
- V33	1	181	1986670	4947.0
- V2	1	358	1986847	4947.1
- V14	1	442	1986931	4947.1
- V30	1	965	1987454	4947.3
- V51	1	1086	1987575	4947.3
- V36	1	1657	1988146	4947.5
- V41	1	1897	1988386	4947.5
- V8	1	2206	1988696	4947.6
- V29	1	2342	1988832	4947.7
- V32	1	2347	1988836	4947.7
- V40	1	2553	1989043	4947.7
- V23	1	3190	1989679	4947.9
- V31	1	3694	1990183	4948.1
- V52	1	3921	1990410	4948.2
- V44	1	4082	1990571	4948.2
- V45	1	4259	1990748	4948.3
- V18	1	4510	1990999	4948.3
- V47	1	6405	1992894	4948.9
<none>			1986489	4949.0
- V39	1	10493	1996982	4950.1
- V26	1	13433	1999922	4951.0
- V3	1	13531	2000020	4951.0
- V10	1	15136	2001625	4951.5
- V20	1	20128	2006617	4953.0

```

- V11 1 25578 2012067 4954.6
- V1 1 25894 2012384 4954.7
- V13 1 30156 2016645 4956.0
- V35 1 34170 2020659 4957.2
- V28 1 34909 2021398 4957.4
- V22 1 53692 2040181 4963.0
- V7 1 54078 2040567 4963.1
- V25 1 65304 2051793 4966.4
- V15 1 83626 2070115 4971.7
- V24 1 95011 2081500 4975.0
- V9 1 99520 2086009 4976.3
- V6 1 142180 2128669 4988.4
- V21 1 144457 2130946 4989.1
- V16 1 173760 2160249 4997.3
- V17 1 177582 2164071 4998.3
- V19 1 358417 2344907 5046.5

```

Step: AIC=4946.99

```

V54 ~ V1 + V2 + V3 + V4 + V6 + V7 + V8 + V9 + V10 + V11 + V13 +
      V14 + V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 + V23 +
      V24 + V25 + V26 + V28 + V29 + V30 + V31 + V32 + V33 + V35 +
      V36 + V39 + V40 + V41 + V44 + V45 + V47 + V48 + V51 + V52

```

	Df	Sum of Sq	RSS	AIC
- V4	1	145	1986701	4945.0
- V48	1	149	1986705	4945.0
- V2	1	355	1986911	4945.1
- V33	1	388	1986944	4945.1
- V14	1	944	1987500	4945.3
- V51	1	1101	1987657	4945.3
- V30	1	1439	1987995	4945.4
- V36	1	1618	1988174	4945.5
- V41	1	1887	1988443	4945.6
- V8	1	2140	1988696	4945.6
- V32	1	2287	1988843	4945.7
- V40	1	2569	1989126	4945.8
- V29	1	2750	1989307	4945.8
- V23	1	3123	1989679	4945.9
- V52	1	3953	1990509	4946.2
- V44	1	4071	1990627	4946.2
- V45	1	4239	1990796	4946.3
- V31	1	4489	1991045	4946.3
- V18	1	4504	1991060	4946.3
- V47	1	6338	1992894	4946.9
<none>			1986556	4947.0
- V39	1	10433	1996989	4948.1
- V26	1	13377	1999933	4949.0
- V3	1	14336	2000893	4949.3
- V10	1	17405	2003961	4950.2
- V20	1	20384	2006940	4951.1
- V1	1	26464	2013020	4952.9
- V11	1	26556	2013112	4953.0
- V35	1	34194	2020750	4955.2
- V13	1	34932	2021488	4955.4
- V28	1	35081	2021637	4955.5
- V7	1	59570	2046126	4962.7

```

- V22 1 63337 2049894 4963.8
- V25 1 66700 2053256 4964.8
- V15 1 84926 2071482 4970.1
- V24 1 97031 2083587 4973.6
- V9 1 100221 2086778 4974.5
- V6 1 142188 2128744 4986.5
- V21 1 145359 2131915 4987.4
- V16 1 177183 2163739 4996.3
- V17 1 179077 2165633 4996.8
- V19 1 360717 2347273 5045.1

```

Step: AIC=4945.03

```

V54 ~ V1 + V2 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V14 +
      V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 +
      V25 + V26 + V28 + V29 + V30 + V31 + V32 + V33 + V35 + V36 +
      V39 + V40 + V41 + V44 + V45 + V47 + V48 + V51 + V52

```

	Df	Sum of Sq	RSS	AIC
- V48	1	172	1986873	4943.1
- V2	1	392	1987093	4943.2
- V33	1	414	1987115	4943.2
- V14	1	944	1987645	4943.3
- V51	1	1086	1987787	4943.4
- V30	1	1495	1988196	4943.5
- V36	1	1676	1988377	4943.5
- V41	1	1939	1988640	4943.6
- V8	1	2061	1988762	4943.7
- V32	1	2327	1989028	4943.7
- V40	1	2640	1989341	4943.8
- V29	1	2822	1989523	4943.9
- V23	1	3028	1989729	4943.9
- V52	1	3865	1990566	4944.2
- V44	1	4137	1990838	4944.3
- V45	1	4303	1991004	4944.3
- V31	1	4480	1991181	4944.4
- V18	1	4532	1991233	4944.4
- V47	1	6343	1993044	4944.9
<none>			1986701	4945.0
- V39	1	10710	1997411	4946.3
- V26	1	13347	2000048	4947.1
- V3	1	14322	2001023	4947.3
- V10	1	17312	2004013	4948.2
- V20	1	20659	2007360	4949.2
- V1	1	26399	2013100	4951.0
- V11	1	26585	2013286	4951.0
- V35	1	34263	2020964	4953.3
- V13	1	34856	2021557	4953.5
- V28	1	35146	2021847	4953.6
- V7	1	59725	2046426	4960.8
- V22	1	63468	2050169	4961.9
- V25	1	66743	2053444	4962.9
- V15	1	84909	2071610	4968.1
- V24	1	97439	2084140	4971.8
- V9	1	100572	2087273	4972.7
- V6	1	142174	2128875	4984.5
- V21	1	145389	2132090	4985.4

```
- V16 1 177392 2164093 4994.3
- V17 1 180113 2166814 4995.1
- V19 1 362162 2348863 5043.5
```

Step: AIC=4943.09

```
V54 ~ V1 + V2 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V14 +
      V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 +
      V25 + V26 + V28 + V29 + V30 + V31 + V32 + V33 + V35 + V36 +
      V39 + V40 + V41 + V44 + V45 + V47 + V51 + V52
```

	Df	Sum of Sq	RSS	AIC
- V33	1	408	1987280	4941.2
- V2	1	417	1987289	4941.2
- V51	1	923	1987796	4941.4
- V14	1	949	1987822	4941.4
- V30	1	1478	1988350	4941.5
- V36	1	1630	1988503	4941.6
- V8	1	1944	1988816	4941.7
- V32	1	2308	1989181	4941.8
- V29	1	2854	1989726	4941.9
- V23	1	2891	1989764	4942.0
- V52	1	3708	1990580	4942.2
- V31	1	4537	1991410	4942.5
- V18	1	4588	1991460	4942.5
- V45	1	4662	1991534	4942.5
- V40	1	4708	1991580	4942.5
- V44	1	4715	1991587	4942.5
- V41	1	5289	1992161	4942.7
<none>			1986873	4943.1
- V47	1	11064	1997936	4944.4
- V39	1	11157	1998030	4944.4
- V26	1	13603	2000476	4945.2
- V3	1	14199	2001072	4945.4
- V10	1	17200	2004072	4946.3
- V20	1	20875	2007747	4947.4
- V1	1	26438	2013310	4949.0
- V11	1	26847	2013720	4949.1
- V13	1	34706	2021579	4951.5
- V28	1	35028	2021901	4951.6
- V35	1	48607	2035480	4955.6
- V7	1	60454	2047326	4959.1
- V22	1	64108	2050980	4960.1
- V25	1	67141	2054014	4961.0
- V15	1	85157	2072030	4966.3
- V24	1	97424	2084296	4969.8
- V9	1	100581	2087454	4970.7
- V6	1	142180	2129053	4982.6
- V21	1	145393	2132266	4983.5
- V16	1	177627	2164499	4992.5
- V17	1	182580	2169453	4993.8
- V19	1	362960	2349833	5041.8

Step: AIC=4941.21

```
V54 ~ V1 + V2 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V14 +
      V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 +
      V25 + V26 + V28 + V29 + V30 + V31 + V32 + V35 + V36 + V39 +
```


V40 + V41 + V44 + V45 + V47 + V51 + V52

	Df	Sum of Sq	RSS	AIC
- V2	1	444	1987724	4939.3
- V51	1	1028	1988308	4939.5
- V36	1	1562	1988842	4939.7
- V8	1	1637	1988918	4939.7
- V14	1	1739	1989019	4939.7
- V32	1	2094	1989375	4939.8
- V23	1	2552	1989832	4940.0
- V29	1	3524	1990804	4940.3
- V52	1	3964	1991244	4940.4
- V31	1	4137	1991417	4940.5
- V18	1	4274	1991554	4940.5
- V30	1	4402	1991682	4940.5
- V40	1	4602	1991882	4940.6
- V44	1	4923	1992204	4940.7
- V45	1	4953	1992233	4940.7
- V41	1	5347	1992627	4940.8
<none>			1987280	4941.2
- V39	1	10863	1998144	4942.5
- V47	1	11249	1998529	4942.6
- V26	1	13263	2000543	4943.2
- V3	1	15453	2002733	4943.9
- V10	1	18253	2005533	4944.7
- V20	1	20765	2008046	4945.4
- V1	1	26979	2014259	4947.3
- V11	1	27893	2015173	4947.6
- V28	1	34729	2022009	4949.6
- V13	1	35327	2022607	4949.8
- V35	1	50035	2037315	4954.1
- V25	1	71718	2058998	4960.5
- V7	1	79762	2067042	4962.8
- V15	1	84873	2072153	4964.3
- V22	1	86050	2073330	4964.6
- V24	1	98413	2085693	4968.2
- V9	1	102954	2090234	4969.5
- V6	1	144775	2132055	4981.4
- V21	1	145331	2132611	4981.6
- V17	1	183472	2170752	4992.2
- V16	1	184024	2171304	4992.3
- V19	1	363592	2350872	5040.0

Step: AIC=4939.34

V54 ~ V1 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V14 + V15 +
V16 + V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 + V25 +
V26 + V28 + V29 + V30 + V31 + V32 + V35 + V36 + V39 + V40 +
V41 + V44 + V45 + V47 + V51 + V52

	Df	Sum of Sq	RSS	AIC
- V51	1	907	1988631	4937.6
- V8	1	1557	1989281	4937.8
- V36	1	1638	1989362	4937.8
- V14	1	1706	1989430	4937.9
- V32	1	1998	1989722	4937.9
- V23	1	2463	1990187	4938.1

```

- V29 1 3521 1991245 4938.4
- V52 1 3765 1991489 4938.5
- V31 1 4273 1991997 4938.6
- V30 1 4274 1991998 4938.6
- V18 1 4338 1992062 4938.7
- V44 1 4682 1992406 4938.8
- V45 1 4739 1992463 4938.8
- V40 1 4743 1992467 4938.8
- V41 1 5726 1993450 4939.1
<none> 1987724 4939.3
- V47 1 11063 1998787 4940.7
- V39 1 11112 1998836 4940.7
- V26 1 13374 2001098 4941.4
- V3 1 15189 2002913 4941.9
- V10 1 18038 2005762 4942.8
- V20 1 20987 2008711 4943.6
- V1 1 26907 2014631 4945.4
- V11 1 28040 2015764 4945.7
- V28 1 34862 2022586 4947.8
- V13 1 35288 2023012 4947.9
- V35 1 49591 2037316 4952.1
- V25 1 72109 2059833 4958.7
- V7 1 79430 2067154 4960.9
- V15 1 84536 2072260 4962.3
- V22 1 85674 2073398 4962.7
- V24 1 98057 2085781 4966.2
- V9 1 102565 2090289 4967.5
- V6 1 144344 2132068 4979.4
- V21 1 144887 2132611 4979.6
- V17 1 183725 2171449 4990.4
- V16 1 184098 2171822 4990.5
- V19 1 363150 2350874 5038.0

```

Step: AIC=4937.62

```

V54 ~ V1 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V14 + V15 +
      V16 + V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 + V25 +
      V26 + V28 + V29 + V30 + V31 + V32 + V35 + V36 + V39 + V40 +
      V41 + V44 + V45 + V47 + V52

```

	Df	Sum of Sq	RSS	AIC
- V14	1	1602	1990234	4936.1
- V8	1	1616	1990248	4936.1
- V36	1	1642	1990273	4936.1
- V32	1	1874	1990505	4936.2
- V23	1	2535	1991167	4936.4
- V52	1	3310	1991942	4936.6
- V29	1	3358	1991989	4936.6
- V45	1	3833	1992464	4936.8
- V30	1	4142	1992773	4936.9
- V18	1	4384	1993016	4936.9
- V31	1	4419	1993051	4936.9
- V44	1	4801	1993432	4937.1
- V40	1	6094	1994725	4937.5
<none>			1988631	4937.6
- V41	1	7889	1996521	4938.0
- V47	1	10947	1999579	4938.9

```

- V39 1 13646 2002278 4939.7
- V26 1 14043 2002674 4939.8
- V3 1 15370 2004002 4940.2
- V10 1 18124 2006755 4941.1
- V20 1 21181 2009813 4942.0
- V1 1 27179 2015810 4943.8
- V11 1 29244 2017875 4944.4
- V28 1 35442 2024073 4946.2
- V13 1 36022 2024653 4946.4
- V25 1 71684 2060316 4956.9
- V35 1 74181 2062813 4957.6
- V7 1 78865 2067497 4959.0
- V15 1 84781 2073412 4960.7
- V22 1 85152 2073783 4960.8
- V24 1 97341 2085972 4964.3
- V9 1 101771 2090402 4965.6
- V6 1 143549 2132181 4977.4
- V21 1 144328 2132959 4977.7
- V16 1 183658 2172289 4988.6
- V17 1 185088 2173720 4989.0
- V19 1 364172 2352803 5036.5

```

Step: AIC=4936.1

```

V54 ~ V1 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 + V25 + V26 +
      V28 + V29 + V30 + V31 + V32 + V35 + V36 + V39 + V40 + V41 +
      V44 + V45 + V47 + V52

```

	Df	Sum of Sq	RSS	AIC
- V32	1	1099	1991333	4934.4
- V36	1	1778	1992012	4934.6
- V29	1	1976	1992210	4934.7
- V8	1	2298	1992532	4934.8
- V30	1	3174	1993407	4935.1
- V23	1	3255	1993489	4935.1
- V52	1	3267	1993501	4935.1
- V45	1	3869	1994103	4935.3
- V44	1	4897	1995131	4935.6
- V18	1	5837	1996071	4935.9
- V31	1	6071	1996305	4935.9
- V40	1	6193	1996427	4936.0
<none>			1990234	4936.1
- V41	1	8428	1998662	4936.6
- V47	1	10625	2000859	4937.3
- V39	1	13778	2004012	4938.2
- V3	1	14269	2004503	4938.4
- V10	1	16605	2006839	4939.1
- V26	1	18043	2008277	4939.5
- V20	1	25346	2015580	4941.7
- V1	1	27630	2017864	4942.4
- V28	1	35558	2025792	4944.7
- V13	1	36351	2026585	4945.0
- V11	1	52368	2042601	4949.7
- V35	1	73857	2064091	4956.0
- V7	1	82153	2072387	4958.4
- V15	1	85835	2076069	4959.4

```

- V22 1 89139 2079373 4960.4
- V25 1 90699 2080933 4960.8
- V9 1 114624 2104858 4967.7
- V24 1 119235 2109469 4969.0
- V6 1 159839 2150072 4980.4
- V21 1 183683 2173917 4987.1
- V17 1 209801 2200035 4994.2
- V16 1 235429 2225663 5001.2
- V19 1 385763 2375997 5040.4

```

Step: AIC=4934.43

```

V54 ~ V1 + V3 + V6 + V7 + V8 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 + V25 + V26 +
      V28 + V29 + V30 + V31 + V35 + V36 + V39 + V40 + V41 + V44 +
      V45 + V47 + V52

```

	Df	Sum of Sq	RSS	AIC
- V8	1	1406	1992739	4932.9
- V36	1	1822	1993155	4933.0
- V23	1	2233	1993566	4933.1
- V29	1	3149	1994482	4933.4
- V52	1	3443	1994776	4933.5
- V45	1	4462	1995795	4933.8
- V44	1	5221	1996554	4934.0
- V40	1	6435	1997768	4934.4
<none>			1991333	4934.4
- V18	1	6946	1998279	4934.5
- V41	1	9018	2000351	4935.1
- V30	1	9353	2000686	4935.2
- V47	1	11474	2002807	4935.9
- V39	1	14127	2005460	4936.7
- V3	1	14524	2005857	4936.8
- V10	1	16372	2007705	4937.3
- V26	1	17196	2008529	4937.6
- V1	1	28094	2019427	4940.8
- V20	1	30747	2022080	4941.6
- V13	1	35337	2026670	4943.0
- V28	1	36953	2028286	4943.5
- V11	1	51289	2042622	4947.7
- V35	1	72778	2064111	4954.0
- V7	1	82632	2073965	4956.8
- V15	1	85326	2076659	4957.6
- V22	1	89140	2080473	4958.7
- V25	1	102311	2093644	4962.5
- V9	1	113898	2105231	4965.8
- V24	1	118659	2109992	4967.2
- V31	1	128759	2120092	4970.0
- V6	1	158879	2150212	4978.5
- V21	1	182969	2174302	4985.2
- V17	1	243929	2235262	5001.8
- V16	1	260264	2251597	5006.1
- V19	1	410901	2402234	5045.0

Step: AIC=4932.85

```

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 + V25 + V26 +

```

V28 + V29 + V30 + V31 + V35 + V36 + V39 + V40 + V41 + V44 +
V45 + V47 + V52

	Df	Sum of Sq	RSS	AIC
- V36	1	1789	1994529	4931.4
- V23	1	2856	1995595	4931.7
- V52	1	3456	1996196	4931.9
- V45	1	4571	1997311	4932.2
- V29	1	4623	1997362	4932.2
- V44	1	5201	1997940	4932.4
<none>			1992739	4932.9
- V40	1	6804	1999544	4932.9
- V30	1	8416	2001155	4933.4
- V41	1	8904	2001644	4933.5
- V47	1	10779	2003518	4934.1
- V18	1	10894	2003633	4934.1
- V3	1	13256	2005995	4934.8
- V39	1	14106	2006845	4935.1
- V10	1	15032	2007771	4935.4
- V26	1	17357	2010096	4936.1
- V1	1	26712	2019451	4938.8
- V13	1	33932	2026672	4941.0
- V28	1	35793	2028532	4941.5
- V20	1	42747	2035486	4943.6
- V11	1	52306	2045046	4946.4
- V35	1	72412	2065152	4952.3
- V15	1	83942	2076681	4955.6
- V7	1	95911	2088650	4959.1
- V22	1	100685	2093424	4960.4
- V25	1	106346	2099085	4962.0
- V9	1	120615	2113354	4966.1
- V24	1	124951	2117690	4967.3
- V31	1	133876	2126615	4969.9
- V6	1	160805	2153545	4977.4
- V21	1	184749	2177489	4984.1
- V16	1	266978	2259718	5006.3
- V17	1	282111	2274850	5010.3
- V19	1	437507	2430246	5049.9

Step: AIC=4931.39

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
V17 + V18 + V19 + V20 + V21 + V22 + V23 + V24 + V25 + V26 +
V28 + V29 + V30 + V31 + V35 + V39 + V40 + V41 + V44 + V45 +
V47 + V52

	Df	Sum of Sq	RSS	AIC
- V23	1	2926	1997455	4930.3
- V52	1	3526	1998054	4930.5
- V29	1	4617	1999146	4930.8
- V45	1	4728	1999257	4930.8
- V44	1	5818	2000347	4931.1
- V40	1	6567	2001096	4931.4
<none>			1994529	4931.4
- V30	1	8165	2002694	4931.8
- V41	1	8631	2003160	4932.0
- V47	1	10901	2005430	4932.7

```

- V18 1 11087 2005616 4932.7
- V3 1 12703 2007232 4933.2
- V39 1 13493 2008022 4933.4
- V10 1 15028 2009557 4933.9
- V26 1 17308 2011837 4934.6
- V1 1 26456 2020985 4937.3
- V13 1 33380 2027909 4939.4
- V28 1 35072 2029601 4939.9
- V20 1 42129 2036658 4941.9
- V11 1 52194 2046722 4944.9
- V35 1 73132 2067661 4951.0
- V15 1 84268 2078797 4954.2
- V7 1 96402 2090931 4957.7
- V22 1 101167 2095696 4959.1
- V25 1 107275 2101804 4960.8
- V9 1 121242 2115771 4964.8
- V24 1 125570 2120098 4966.0
- V31 1 134840 2129368 4968.6
- V6 1 160981 2155510 4976.0
- V21 1 184799 2179328 4982.6
- V16 1 268065 2262593 5005.1
- V17 1 280900 2275429 5008.4
- V19 1 436666 2431195 5048.2

```

Step: AIC=4930.27

```

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
      V29 + V30 + V31 + V35 + V39 + V40 + V41 + V44 + V45 + V47 +
      V52

```

	Df	Sum of Sq	RSS	AIC
- V52	1	3715	2001170	4929.4
- V45	1	4678	2002133	4929.7
- V29	1	5491	2002946	4929.9
- V44	1	6017	2003472	4930.1
- V40	1	6453	2003908	4930.2
<none>			1997455	4930.3
- V30	1	8053	2005508	4930.7
- V18	1	8460	2005915	4930.8
- V41	1	8800	2006255	4930.9
- V47	1	10832	2008287	4931.5
- V10	1	12889	2010344	4932.1
- V39	1	13832	2011287	4932.4
- V26	1	15844	2013299	4933.0
- V3	1	17340	2014795	4933.5
- V1	1	27721	2025176	4936.5
- V13	1	30631	2028085	4937.4
- V28	1	37141	2034596	4939.3
- V20	1	45547	2043002	4941.8
- V11	1	50870	2048325	4943.4
- V35	1	75958	2073413	4950.7
- V25	1	104913	2102368	4959.0
- V7	1	114544	2111999	4961.7
- V22	1	124854	2122309	4964.7
- V9	1	125834	2123289	4964.9
- V15	1	130450	2127905	4966.2

```

- V24 1 132509 2129964 4966.8
- V31 1 135477 2132932 4967.6
- V6 1 159502 2156957 4974.4
- V21 1 184991 2182446 4981.4
- V16 1 268786 2266241 5004.0
- V17 1 338051 2335506 5022.1
- V19 1 497873 2495327 5061.8

```

Step: AIC=4929.39

```

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
      V29 + V30 + V31 + V35 + V39 + V40 + V41 + V44 + V45 + V47

```

	Df	Sum of Sq	RSS	AIC
- V45	1	2615	2003785	4928.2
- V44	1	3703	2004873	4928.5
- V29	1	5507	2006677	4929.0
<none>			2001170	4929.4
- V30	1	7889	2009059	4929.7
- V40	1	7977	2009146	4929.8
- V18	1	8791	2009961	4930.0
- V41	1	12413	2013583	4931.1
- V47	1	12454	2013624	4931.1
- V10	1	13201	2014371	4931.3
- V3	1	16440	2017610	4932.3
- V26	1	16822	2017991	4932.4
- V39	1	19777	2020947	4933.3
- V1	1	26263	2027432	4935.2
- V13	1	30160	2031329	4936.4
- V28	1	36535	2037704	4938.2
- V20	1	46188	2047358	4941.1
- V11	1	53120	2054290	4943.1
- V35	1	72834	2074003	4948.8
- V25	1	105726	2106896	4958.3
- V7	1	114465	2115635	4960.8
- V22	1	124946	2126116	4963.7
- V9	1	125734	2126903	4963.9
- V15	1	128269	2129439	4964.7
- V24	1	132622	2133791	4965.9
- V31	1	138088	2139258	4967.4
- V6	1	160550	2161719	4973.7
- V21	1	187023	2188193	4981.0
- V16	1	268933	2270103	5003.0
- V17	1	336604	2337773	5020.7
- V19	1	497503	2498672	5060.6

Step: AIC=4928.17

```

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
      V29 + V30 + V31 + V35 + V39 + V40 + V41 + V44 + V47

```

	Df	Sum of Sq	RSS	AIC
- V44	1	1382	2005167	4926.6
- V29	1	5698	2009483	4927.9
- V40	1	5768	2009552	4927.9
<none>			2003785	4928.2

```

- V18      1      8487 2012272 4928.7
- V30      1      8623 2012408 4928.7
- V41      1      9915 2013700 4929.1
- V47      1     10100 2013884 4929.2
- V10      1     12872 2016657 4930.0
- V26      1     17440 2021225 4931.4
- V3       1     17834 2021619 4931.5
- V39      1     20628 2024413 4932.3
- V1       1     28538 2032322 4934.7
- V13      1     32097 2035881 4935.7
- V28      1     38786 2042571 4937.7
- V20      1     47511 2051295 4940.2
- V11      1     54132 2057917 4942.2
- V35      1     70631 2074416 4947.0
- V25      1    106082 2109867 4957.1
- V7       1    111898 2115683 4958.8
- V22      1    122354 2126139 4961.7
- V9       1    123324 2127109 4962.0
- V15      1    126328 2130113 4962.9
- V24      1    130246 2134031 4964.0
- V31      1    135828 2139612 4965.5
- V6       1    158474 2162259 4971.8
- V21      1    184916 2188700 4979.1
- V16      1    269260 2273044 5001.8
- V17      1    336157 2339941 5019.2
- V19      1    495935 2499719 5058.9

```

Step: AIC=4926.58

```

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
      V29 + V30 + V31 + V35 + V39 + V40 + V41 + V47

```

	Df	Sum of Sq	RSS	AIC
- V40	1	4990	2010157	4926.1
- V29	1	5527	2010694	4926.2
<none>			2005167	4926.6
- V18	1	8319	2013486	4927.1
- V30	1	8646	2013812	4927.2
- V41	1	8670	2013837	4927.2
- V47	1	8721	2013887	4927.2
- V10	1	13230	2018397	4928.5
- V26	1	17214	2022381	4929.7
- V3	1	17909	2023075	4929.9
- V39	1	21128	2026294	4930.9
- V1	1	29448	2034615	4933.3
- V13	1	32393	2037560	4934.2
- V28	1	38900	2044067	4936.1
- V20	1	47159	2052326	4938.5
- V11	1	53978	2059145	4940.5
- V35	1	69732	2074898	4945.1
- V25	1	104879	2110045	4955.2
- V7	1	111686	2116853	4957.1
- V22	1	122191	2127357	4960.1
- V9	1	123056	2128223	4960.3
- V15	1	127483	2132650	4961.6
- V24	1	129911	2135078	4962.3


```

- V31 1 135354 2140521 4963.8
- V6 1 160367 2165534 4970.7
- V21 1 187028 2192195 4978.1
- V16 1 268241 2273408 4999.9
- V17 1 336107 2341274 5017.6
- V19 1 496421 2501587 5057.3

```

Step: AIC=4926.08

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
V29 + V30 + V31 + V35 + V39 + V41 + V47

	Df	Sum of Sq	RSS	AIC
- V29	1	5213	2015371	4925.6
- V41	1	5873	2016030	4925.8
<none>			2010157	4926.1
- V30	1	8334	2018491	4926.6
- V18	1	8589	2018747	4926.6
- V10	1	12419	2022576	4927.8
- V47	1	14896	2025053	4928.5
- V3	1	16930	2027088	4929.1
- V26	1	17125	2027283	4929.2
- V39	1	17397	2027554	4929.2
- V1	1	25870	2036027	4931.7
- V13	1	31784	2041941	4933.5
- V28	1	38351	2048508	4935.4
- V20	1	48401	2058558	4938.4
- V11	1	54318	2064475	4940.1
- V35	1	91910	2102067	4950.9
- V25	1	107379	2117536	4955.3
- V7	1	112438	2122595	4956.7
- V9	1	122641	2132799	4959.6
- V22	1	123016	2133173	4959.7
- V15	1	126882	2137039	4960.8
- V24	1	129481	2139638	4961.5
- V31	1	140675	2150832	4964.7
- V6	1	159856	2170013	4970.0
- V21	1	186958	2197115	4977.4
- V16	1	269336	2279493	4999.5
- V17	1	334886	2345043	5016.5
- V19	1	493847	2504004	5055.9

Step: AIC=4925.63

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
V30 + V31 + V35 + V39 + V41 + V47

	Df	Sum of Sq	RSS	AIC
- V41	1	5772	2021143	4925.3
<none>			2015371	4925.6
- V30	1	7284	2022655	4925.8
- V18	1	10522	2025892	4926.8
- V10	1	12409	2027779	4927.3
- V47	1	14614	2029985	4928.0
- V39	1	17900	2033271	4928.9
- V3	1	22154	2037525	4930.2

```

- V26 1 30570 2045941 4932.7
- V13 1 30740 2046110 4932.7
- V1 1 32437 2047808 4933.2
- V28 1 37219 2052590 4934.6
- V20 1 43400 2058771 4936.4
- V11 1 65710 2081081 4942.9
- V35 1 89647 2105018 4949.7
- V25 1 103626 2118996 4953.7
- V7 1 108274 2123645 4955.0
- V22 1 119084 2134454 4958.1
- V9 1 120996 2136366 4958.6
- V24 1 131807 2147178 4961.6
- V15 1 138173 2153544 4963.4
- V31 1 145604 2160975 4965.5
- V6 1 155387 2170758 4968.2
- V21 1 183940 2199311 4976.0
- V16 1 276782 2292153 5000.8
- V17 1 330484 2345854 5014.7
- V19 1 503793 2519164 5057.5

```

Step: AIC=4925.35

```

V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 + V15 + V16 +
      V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 + V26 + V28 +
      V30 + V31 + V35 + V39 + V47

```

	Df	Sum of Sq	RSS	AIC
<none>			2021143	4925.3
- V30	1	7385	2028528	4925.5
- V18	1	10310	2031453	4926.4
- V47	1	11537	2032679	4926.8
- V10	1	11792	2032934	4926.8
- V39	1	12995	2034138	4927.2
- V3	1	22300	2043443	4929.9
- V13	1	29930	2051072	4932.2
- V26	1	32236	2053378	4932.8
- V1	1	33951	2055094	4933.3
- V28	1	37060	2058203	4934.2
- V20	1	41152	2062294	4935.4
- V11	1	67419	2088561	4943.0
- V35	1	102399	2123542	4953.0
- V25	1	107682	2128824	4954.5
- V7	1	108638	2129781	4954.8
- V22	1	118949	2140092	4957.7
- V9	1	120467	2141610	4958.1
- V24	1	130870	2152012	4961.0
- V15	1	141810	2162953	4964.0
- V31	1	144247	2165390	4964.7
- V6	1	153136	2174278	4967.2
- V21	1	181346	2202489	4974.9
- V16	1	277497	2298640	5000.5
- V17	1	332115	2353258	5014.6
- V19	1	501259	2522402	5056.3

```
> summary(fitt)
```

Call:

```
lm(formula = V54 ~ V1 + V3 + V6 + V7 + V9 + V10 + V11 + V13 +
```

```

V15 + V16 + V17 + V18 + V19 + V20 + V21 + V22 + V24 + V25 +
V26 + V28 + V30 + V31 + V35 + V39 + V47, data = fbmsg)

Residuals:
    Min       1Q   Median       3Q      Max
-434.45  -18.13   -4.64   10.72  649.34

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.397e+00  6.263e+00   1.021  0.307501
V1          -1.082e-06  3.485e-07  -3.105  0.001996 **
V3           2.141e-05  8.507e-06   2.517  0.012121 *
V6           7.712e-01  1.169e-01   6.595  9.68e-11 ***
V7           1.015e+01  1.828e+00   5.555  4.26e-08 ***
V9          -7.333e+00  1.254e+00  -5.849  8.30e-09 ***
V10          5.600e-01  3.060e-01   1.830  0.067770 .
V11          2.641e-01  6.036e-02   4.376  1.44e-05 ***
V13          8.796e-01  3.017e-01   2.915  0.003690 **
V15          2.012e+00  3.170e-01   6.346  4.48e-10 ***
V16         -4.482e-01  5.049e-02  -8.877 < 2e-16 ***
V17         -4.222e+00  4.347e-01  -9.712 < 2e-16 ***
V18          4.490e-01  2.624e-01   1.711  0.087594 .
V19          3.241e+00  2.717e-01  11.931 < 2e-16 ***
V20          6.203e-01  1.814e-01   3.419  0.000674 ***
V21         -9.018e-01  1.257e-01  -7.176  2.22e-12 ***
V22         -1.080e+01  1.858e+00  -5.812  1.02e-08 ***
V24          7.865e+00  1.290e+00   6.096  1.99e-09 ***
V25         -2.470e-01  4.467e-02  -5.530  4.87e-08 ***
V26         -1.567e-01  5.179e-02  -3.026  0.002592 **
V28         -9.408e-01  2.900e-01  -3.244  0.001246 **
V30          3.182e-02  2.197e-02   1.448  0.148104
V31          1.726e-01  2.697e-02   6.400  3.22e-10 ***
V35         -8.723e-01  1.618e-01  -5.393  1.02e-07 ***
V39          5.541e-01  2.884e-01   1.921  0.055214 .
V47          9.882e+00  5.459e+00   1.810  0.070807 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59.34 on 574 degrees of freedom
Multiple R-squared:  0.558,    Adjusted R-squared:  0.5388
F-statistic: 28.99 on 25 and 574 DF,  p-value: < 2.2e-16

> prediction5<-predict(fitt, fbmsg)
> mse5<-mean((fbmsg$V54-prediction5)^2)
> mse5
[1] 3368.571

```

```
library(MASS)
```

```
fit_rgm<-rlm(V54~., data=fbmsg)
```

```
fit_rgm
```

#2. Report the training accuracy and test accuracy

#Lasso model accuracy prediction

```
mse<-mean((y-predictions2)^2)
```

```
mse
```

```
#Lasso model accuracy prediction
> mse<-mean((y-predictions2)^2)
> mse
[1] 3403.824
```

Ridge model accuracy prediction

```
mse2<-mean((fbmsg$V54-predictions3)^2)
```

```
mse2
```

```
# Ridge model accuracy prediction
> mse2<-mean((fbmsg$V54-predictions3)^2)
> mse2
[1] 3406.419
```

stepwise model accuracy

```
mse5<-mean((fbmsg$V54-prediction5)^2)
```

```
mse5
```

```
> # stepwise model accuracy
> mse5<-mean((fbmsg$V54-prediction5)^2)
> mse5
[1] 3368.571
```

#3. compare with linear models and report the accuracy

```
library(lars)
```

```
x<-as.matrix(fbmsg[,1:53])
```

```
y<-as.matrix(fbmsg[,54])
```

```
fit<-lars(x,y,type = "lasso")
```

```
fit
```

```
summary(fit)
```

```
best_step<-fit$df[which.min(fit$RSS)]
```

```
best_step
```

```
predictions6<-predict(fit,x, s=best_step, type = "fit")$fit
```

```
mse6<-mean((y-predictions6)^2)
```

```
mse6
```

```
library(pls)
```

```
fit_pcr<-pcr(V54~., data=fbmsg, validation="CV")
```

```
fit_pcr
```

```
summary(fit_pcr)
```

```
> library(lars)
Loaded lars 1.2

> x<-as.matrix(fbmsg[,1:53])
> y<-as.matrix(fbmsg[,54])
> fit<-lars(x,y,type = "lasso")
> fit

Call:
lars(x = x, y = y, type = "lasso")
R-squared: 0.566
Sequence of LASSO moves:
      V31 V19 V24 V35 V24 V28 V24 V14 V24 V3 V39 V34 V52 V23 V40 V16 V1 V46 V
15 V36 V5 V37 V6 V18 V47
Var   31  19  24  35 -24  28  24  14 -24  3  39  34  52  23  40  16  1  46
15  36  5  37  6  18  47
Step   1   2   3   4   5   6   7   8   9 10 11 12 13 14 15 16 17 18
19  20 21  22 23  24  25
      V45 V29 V41 V4 V53 V21 V10 V9 V6 V27 V17 V25 V29 V2 V13 V27 V3 V11 V51
V9 V40 V3 V22 V29 V26 V34
Var   45  29  41  4  53  21  10  9 -6  27  17  25 -29  2  13 -27 -3  11  51
-9 -40  3  22  29  26 -34
Step  26  27  28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
45  46 47  48 49 50 51
      V6 V30 V9 V44 V34 V53 V12 V33 V2 V24 V48 V8 V2 V7 V45 V40 V20 V12 V27 V
5 V5 V32 V34 V34 V12 V45
Var   6  30  9  44  34 -53  12  33 -2  24  48  8  2  7 -45  40  20 -12  27 -
5  5  32 -34  34 -12  45
Step  52  53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 7
1 72 73 74 75 76 77
      V41 V53 V41 V44 V44
Var  -41 -53  41 -44  44
Step  78  79  80  81  82
> summary(fit)
LARS/LASSO
Call: lars(x = x, y = y, type = "lasso")
      Df      Rss      Cp
0      1 4573220 679.728
1      2 3142406 281.968
```

2	3	3080272	266.608
3	4	3073055	266.592
4	5	3067478	267.034
5	4	2899570	218.121
6	5	2865308	210.549
7	6	2806464	196.108
8	7	2787846	192.906
9	6	2763019	183.970
10	7	2755779	183.947
11	8	2747079	183.516
12	9	2740251	183.608
13	10	2668642	165.601
14	11	2653828	163.463
15	12	2649850	164.351
16	13	2634753	162.133
17	14	2633365	163.745
18	15	2624415	163.245
19	16	2607612	160.550
20	17	2541884	144.186
21	18	2528050	142.321
22	19	2516039	140.965
23	20	2514488	142.532
24	21	2511862	143.798
25	22	2508903	144.971
26	23	2503701	145.518
27	24	2499746	146.413
28	25	2492473	146.381
29	26	2491994	148.247
30	27	2484788	148.234
31	28	2481442	149.299
32	29	2477212	150.117
33	30	2476786	151.998
34	29	2476406	149.892
35	30	2473583	151.103
36	31	2454048	147.645
37	32	2441628	146.175
38	31	2418760	137.786
39	32	2408484	136.915
40	33	2377594	130.285
41	32	2348943	120.279
42	31	2344943	117.162
43	32	2342915	118.596
44	33	2322788	114.972
45	32	2320057	112.209
46	31	2316059	109.092
47	32	2299413	106.441
48	33	2265135	98.864
49	34	2252744	97.402
50	35	2214562	88.734
51	34	2214080	86.600
52	35	2202527	85.372
53	36	2133776	68.163
54	37	2131966	69.658
55	38	2128231	70.614
56	39	2123506	71.294
57	38	2120832	68.547

```

58 39 2101128 65.042
59 40 2099918 66.704
60 39 2096208 63.667
61 40 2093946 65.035
62 41 2090458 66.061
63 42 2078989 64.856
64 43 2061008 61.832
65 44 2017935 51.798
66 43 2012648 48.321
67 44 1994522 45.257
68 45 1988388 45.543
69 44 1987703 43.352
70 45 1987026 45.162
71 44 1986505 43.017
72 45 1986445 45.000
73 46 1986445 47.000
74 45 1986445 45.000
75 46 1986445 47.000
76 45 1986445 45.000
77 46 1986445 47.000
78 45 1986445 45.000
79 44 1986445 43.000
80 45 1986445 45.000
81 44 1986445 43.000
82 45 1986445 45.000
> best_step<-fit$df[which.min(fit$RSS)]
> best_step

45
> predictions6<-predict(fit,x, s=best_step, type = "fit")$fit
> mse6<-mean((y-predictions6)^2)
> mse6
[1] 3871.314

```

```

predictions7<-predict(fit_pcr, fbmsg, ncomp = 6)
mse7<-mean((fbmsg$V54-predictions7)^2)
mse7

```

```

library(pls)
fit_pls<-plsr(V54~., data=fbmsg, validation="CV")
fit_pls
summary(fit_pls)

```

```

package 'pls' was built under R version 3.6.1
> fit_pcr<-pcr(V54~., data=fbmsg, validation="CV")
> fit_pcr
Principal component regression , fitted with the singular value decomposition algorithm.
Cross-validated using 10 random segments.
Call:
pcr(formula = V54 ~ ., data = fbmsg, validation = "CV")

```

```

> summary(fit_pcr)
Data:   X dimension: 600 53
        Y dimension: 600 1
Fit method: svdpc
Number of components considered: 53

VALIDATION: RMSEP
Cross-validated using 10 random segments.
      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7
comps  8 comps  9 comps
CV      87.45   87.55   87.21   87.22   82.82   79.14   75.89
75.22    74.70   73.17
adjCV    87.45   87.54   87.17   87.18   82.76   78.32   74.88
75.11    74.77   73.00
      10 comps 11 comps 12 comps 13 comps 14 comps 15 comps 16 comps
17 comps 18 comps
CV      73.95   74.09   75.95   74.65   75.26   75.91   76.20
      76.68   77.42
adjCV    73.75   73.88   75.72   74.24   74.84   75.43   75.69
      76.11   76.83
      19 comps 20 comps 21 comps 22 comps 23 comps 24 comps 25 comps
26 comps 27 comps
CV      76.39   76.61   77.66   77.88   77.89   80.43   81.27
      81.71   79.59
adjCV    75.73   75.99   76.99   77.22   77.20   79.51   80.32
      80.74   78.54
      28 comps 29 comps 30 comps 31 comps 32 comps 33 comps 34 comps
35 comps 36 comps
CV      80.64   83.94   79.02   77.36   77.68   77.55   80.38
      80.47   80.31
adjCV    79.51   82.73   78.21   76.18   76.63   76.49   79.07
      79.08   78.95
      37 comps 38 comps 39 comps 40 comps 41 comps 42 comps 43 comps
44 comps 45 comps
CV      80.30   80.16   80.55   81.39   81.54   81.47   81.07
      81.04   81.09
adjCV    78.94   78.81   79.18   79.95   80.10   80.03   79.65
      79.62   79.67
      46 comps 47 comps 48 comps 49 comps 50 comps 51 comps 52 comps
53 comps
CV      81.09   81.09   81.09   81.09   81.09   81.09   81.09
      81.09
adjCV    79.64   79.63   79.62   79.62   79.59   79.59   79.58
      79.58

TRAINING: % variance explained
      1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
9 comps 10 comps
X      99.75033 100.000 100.000 100.00 100.00 100.00 100.00 100.0
100.00 100.00
V54    0.01929 1.896 1.896 12.52 22.84 28.21 29.16 30.3
      34.88 35.39
      11 comps 12 comps 13 comps 14 comps 15 comps 16 comps 17 comps 1
8 comps 19 comps
X      100.00 100.00 100.00 100.00 100.00 100.00 100.00 100.00
100.00 100.00

```



```

V54      35.41      36.03      40.38      40.54      41.11      41.67      41.81
  41.89      42.92
    20 comps  21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  2
7 comps  28 comps
X      100.00      100.00      100.00      100.00      100.00      100.00      100.00
  100.00      100.00
V54      43.05      43.25      43.35      43.85      45.68      45.88      46.35
  47.69      47.97
    29 comps  30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  3
6 comps  37 comps
X      100      100.00      100.00      100.00      100.00      100.00      100.00
  100.00      100.00
V54      48      50.23      53.12      53.79      54.18      54.61      55.72
  55.86      55.89
    38 comps  39 comps  40 comps  41 comps  42 comps  43 comps  44 comps  4
5 comps  46 comps
X      100.00      100.00      100.00      100.00      100.00      100.00      100.00
  100.00      100.00
V54      55.98      56.06      56.32      56.32      56.37      56.54      56.56
  56.56      56.62
    47 comps  48 comps  49 comps  50 comps  51 comps  52 comps  53 comps
X      100.00      100.00      100.00      100.00      100.00      100.00      100.00
V54      56.66      56.68      56.68      56.73      56.74      56.75      56.76
> predictions7<-predict(fit_pcr, fbmsg, ncomp = 6)
> mse7<-mean((fbmsg$V54-predictions7)^2)
> mse7
[1] 5471.747

```

```

predictions8<-predict(fit_pls, fbmsg, ncomp = 6)

```

```

mse8<-mean((fbmsg$V54-predictions8)^2)

```

```

mse8

```

```

> fit_pls<-plsr(V54~., data=fbmsg, validation="CV")
> fit_pls
Partial least squares regression , fitted with the kernel algorithm.
Cross-validated using 10 random segments.
Call:
plsr(formula = V54 ~ ., data = fbmsg, validation = "CV")
> summary(fit_pls)
Data:   X dimension: 600 53
        Y dimension: 600 1
Fit method: kernelpls
Number of components considered: 53

VALIDATION: RMSEP
Cross-validated using 10 random segments.
      (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7
comps  8 comps  9 comps
CV      87.45   87.52   86.81   84.07   79.39   73.60   73.21
73.38   75.01   74.72

```

adjCV	87.45	87.51	86.79	83.74	79.36	73.49	73.03	
73.23	74.52	74.37						
10 comps	11 comps	12 comps	13 comps	14 comps	15 comps	16 comps		
17 comps	18 comps							
CV	74.22	74.25	74.33	74.41	75.61	75.53	75.85	
74.64	75.23							
adjCV	73.85	73.92	73.95	74.17	75.03	74.88	75.20	
74.08	74.46							
19 comps	20 comps	21 comps	22 comps	23 comps	24 comps	25 comps		
26 comps	27 comps							
CV	75.97	75.86	76.64	76.21	76.71	77.36	76.93	
78.09	78.03							
adjCV	75.15	74.99	75.70	75.28	75.68	76.28	75.88	
76.98	76.90							
28 comps	29 comps	30 comps	31 comps	32 comps	33 comps	34 comps		
35 comps	36 comps							
CV	78.39	77.46	79.84	80.28	79.81	80.87	82.53	
80.93	81.56							
adjCV	77.23	76.41	78.56	78.97	78.53	79.51	81.04	
79.55	80.13							
37 comps	38 comps	39 comps	40 comps	41 comps	42 comps	43 comps		
44 comps	45 comps							
CV	81.29	81.18	81.16	81.19	81.23	81.28	81.27	
81.27	81.27							
adjCV	79.87	79.77	79.75	79.78	79.82	79.86	79.85	
79.85	79.85							
46 comps	47 comps	48 comps	49 comps	50 comps	51 comps	52 comps		
53 comps								
CV	81.27	81.27	81.27	81.27	81.27	81.27	81.27	
81.27								
adjCV	79.85	79.85	79.85	79.85	79.85	79.85	79.85	
79.85								
TRAINING: % variance explained								
1 comps	2 comps	3 comps	4 comps	5 comps	6 comps	7 comps	8 comps	
9 comps	10 comps							
X	99.68993	100.000	100.000	100.00	100.00	100.00	100.00	100.00
100.00	100.00							
V54	0.02979	1.896	9.539	19.32	33.42	36.38	37.02	39.59
40.61	41.34							
11 comps	12 comps	13 comps	14 comps	15 comps	16 comps	17 comps	1	
8 comps	19 comps							
X	100.00	100.00	100.00	100.00	100.00	100.00	100.00	
100.00	100.00							
V54	41.75	42.42	42.98	44.61	46.49	47.99	49.01	
50.29	51.24							
20 comps	21 comps	22 comps	23 comps	24 comps	25 comps	26 comps	2	
7 comps	28 comps							
X	100.00	100.00	100.00	100.00	100.00	100.00	100.00	
100.00	100.00							
V54	52.36	52.91	53.39	53.89	54.12	54.23	54.31	
54.46	54.67							
29 comps	30 comps	31 comps	32 comps	33 comps	34 comps	35 comps	3	
6 comps	37 comps							
X	100.00	100.00	100.00	100.00	100.0	100.00	100.00	
100.00	100.00							

```

V54      54.87      55.13      55.18      55.47      55.8      56.06      56.33
      56.49      56.53
      38 comps  39 comps  40 comps  41 comps  42 comps  43 comps  44 comps  4
5 comps  46 comps
X      100.00      100.00      100.00      100.00      100.00      100.00      100.00
      100.00      100.00
V54      56.55      56.55      56.55      56.56      56.56      56.56      56.56
      56.56      56.56
      47 comps  48 comps  49 comps  50 comps  51 comps  52 comps  53 comps
X      100.00      100.00      100.00      100.00      100.00      100.00      100.00
V54      56.56      56.56      56.56      56.56      56.56      56.56      56.56
> predictions8<-predict(fit_pls, fbmsg, ncomp = 6)
> mse8<-mean((fbmsg$V54-predictions8)^2)
> mse8
[1] 4849.094

```

#4. create a graph displaying the accuracy of all models

For plotting

```
plot(fit_lasso, xvar="lambda")
```

```
plot(fit10, main="LASSO")
```

```
plot(fit_ridge, xvar="lambda")
```

```
plot(fit0, main="Ridge")
```

```
plot(fit_elnet, xvar="lambda")
```

```
plot(fit5, main="Elastic Net")
```

```

> plot(fit_lasso, xvar="lambda")
Error in plot.new() : figure margins too large
> plot(fit10, main="LASSO")
Error in plot.new() : figure margins too large
> plot(fit_ridge, xvar="lambda")
Error in plot.new() : figure margins too large
> plot(fit0, main="Ridge")
Error in plot.new() : figure margins too large
> plot(fit_elnet, xvar="lambda")
Error in plot.new() : figure margins too large
> plot(fit5, main="Elastic Net")
Error in plot.new() : figure margins too large

```

#####Sample 2

```
lib=c("bigmemory", "readr", "Hmisc", "dplyr", "MASS", "ggplot2", "lattice", "caret", "rpart",
      "randomForest", "rpart.plot", "lattice", "rattle", "data.table", "RColorBrewer", "reshape2",
      "InformationValue", "stringr", "VIF", "Information", "Amelia", "gdata", "party", "car",
      "lubridate", "zoo", "sqldf", "fuzzyjoin", "party", "mice", "mlbench")
supply(lib, require, character.only=TRUE, quietly=TRUE)
```

```
# import train data set
```

```
data1<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_1.csv")
data2<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_2.csv")
data3<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_3.csv")
data4<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_4.csv")
data5<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_5.csv")
```

```
fbtrain<-rbind(data1,data2,data3,data4,data5)
dim(fbtrain)
```

```
> data1<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_1.csv")
> data2<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_2.csv")
> data3<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_3.csv")
> data4<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_4.csv")
> data5<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_5.csv")
> fbtrain<-rbind(data1,data2,data3,data4,data5)
> dim(fbtrain)
[1] 602813      54
```

```
#import test data set
```

```
test1<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_1.csv")
test2<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_2.csv")
test3<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_3.csv")
test4<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_4.csv")
test5<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_5.csv")
test6<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_6.csv")
test7<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_7.csv")
test8<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_8.csv")
test9<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_9.csv")
test10<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_10.csv")
```

```
#test10<-fread("F:/AcadGild/workings/Dataset/Dataset/Testing/TestSet/Test_Case_10.csv",
header = FALSE)
```

```
fbtest<-rbind(test1,test2,test3,test4,test5,test6,test7,test8,test9,test10)
dim(fbtest)
```

```
test1<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_1.csv")
> test2<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_2.csv")
> test3<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_3.csv")
> test4<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_4.csv")
```

```

> test5<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_5.csv")
> test6<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_6.csv")
> test7<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_7.csv")
> test8<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_8.csv")
> test9<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_9.csv")
> test10<-fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Testing/TestSet/Test_Case_10.csv")
> fbtest<-rbind(test1,test2,test3,test4,test5,test6,test7,test8,test9,test10)
> dim(fbtest)
[1] 1000    54

```

Assign variable names to the train and test data set

```
colnames(fbtrain) <-
```

```
c("plikes","checkin","talking","category","d5","d6","d7","d8","d9","d10","d11","d12",
```

```
"d13","d14","d15","d16","d17","d18","d19","d20","d21","d22","d23","d24","d25","d26",
```

```
"d27","d28","d29","cc1","cc2","cc3","cc4","cc5","basetime","postlength","postshre",
```

```
"postpromo","Hhrs","sun","mon","tue","wed","thu","fri","sat","basesun","basemon",
  "basetue","basewed","basethu","basefri","basesat","target")
```

```
colnames(fbtest) <-
```

```
c("plikes","checkin","talking","category","d5","d6","d7","d8","d9","d10","d11","d12",
```

```
"d13","d14","d15","d16","d17","d18","d19","d20","d21","d22","d23","d24","d25","d26",
```

```
"d27","d28","d29","cc1","cc2","cc3","cc4","cc5","basetime","postlength","postshre",
```

```
"postpromo","Hhrs","sun","mon","tue","wed","thu","fri","sat","basesun","basemon",
  "basetue","basewed","basethu","basefri","basesat","target")
```

exploratory analysis

```
dim(fbtrain); dim(fbtest)
```

```
str(fbtrain); str(fbtest)
View(fbtrain); View(fbtest)
summary(fbtrain); summary(fbtest)
describe(fbtrain); describe(fbtest)
Amelia::missmap(fbtrain); Amelia::missmap(fbtest) # no missing values
```

```
train<-(fbtrain); test<-(fbtest)
head(train); head(test)
```

```
# removing overlapping observations if any
distinct(train)
dim(train)
distinct(test)
dim(test)
```

```
-----
-----
plikes
      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
602813      0      639      1 1313640 2029019      3319      6991      3
6734 292911 1204214
.90   .95
3337470 5365996

Value      0.00e+00 5.00e+06 1.00e+07 1.50e+07 2.00e+07 2.50e+07 3.00e+07 3.
50e+07 4.00e+07 1.65e+08
Frequency      520895      66030      8284      4327      879      15      1441
706      59      88
Proportion      0.864      0.110      0.014      0.007      0.001      0.000      0.002
0.001      0.000      0.000

Value      4.85e+08
Frequency      89
Proportion      0.000
-----
-----
checkin
      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
602813      0      173      0.759      4673      8853      0      0
0      0      99
.90   .95
2842 27805

lowest :      0      1      2      3      4, highest: 120918 122388 175714 1
85932 186370
-----
-----
```

```

talking
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      507      1      44766      67982      10      33
698      7045      50264
.90      .95
120305      194202

Value      0      50000      100000      150000      200000      250000      300000      400000
450000      500000      550000
Frequency      378239      119287      51911      19874      6819      10693      7847      204
925      2658      1588
Proportion      0.627      0.198      0.086      0.033      0.011      0.018      0.013      0.000
0.002      0.004      0.003

Value      600000      650000      800000      850000      1200000      1250000      1300000      3950000      6
100000
Frequency      1167      134      15      75      730      425      118      89
15
Proportion      0.002      0.000      0.000      0.000      0.001      0.001      0.000      0.000
0.000
-----
category
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      81      0.991      24.25      20.12      3      8
9      18      32
.90      .95
46      68

lowest :      1      2      3      4      5, highest:      96 100 101 105 106
-----
d5
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      141      0.056      0.6475      1.288      0      0
0      0      0
.90      .95
0      0

lowest :      0      1      2      3      4, highest: 1458 1494 1841 1923 2341
-----
d6
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      986      1      477.6      553.9      6      15
47      251      760
.90      .95
1310      1485

lowest :      0      1      2      3      4, highest: 2430 2438 2442 2450 2495

```



```

-----
d7
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      2868      1      55.86      72.48      0.7475      1.6448      5.
4191 23.6474 71.3169
.90      .95
148.4750 199.6468

lowest : 0.000000e+00 1.694915e-02 1.704545e-02 3.436426e-02 5.000000e-02
highest: 1.939400e+03 1.950000e+03 2.031000e+03 2.085000e+03 2.341000e+03
-----

```

```

-----
d8
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      541      0.997      35.35      49.37      0      0
2      12      41
.90      .95
100      132

lowest :      0.0      0.5      1.0      1.5      2.0, highest: 2114.0 2123.0 2239.0 2
341.0 2347.0
-----

```

```

-----
d9
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      2975      1      67.97      81.25      1.227      2.629      8
.032 33.772 103.316
.90      .95
192.179 228.596

lowest :      0.0000000      0.1290809      0.1294407      0.2179449      0.2192469
highest: 743.0916498 747.6927928 762.3577463 795.6776287 1023.7656897
-----

```

```

-----
d10
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      68      0.022      0.0937      0.1871      0      0
0      0      0
.90      .95
0      0

lowest :      0      1      2      3      4, highest: 381 434 727 1311 1923
-----

```

```

-----
d11
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      892      1      360.6      432.1      5      10
38      175      552
.90      .95
1033      1248
-----

```


Value	62	79	113	148	190	324			
Frequency	3	4	5	4	1	1			
Proportion	0	0	0	0	0	0			

d16

	n	missing	distinct	Info	Mean	Gmd	.05	.10
.25		.50	.75					
602813		0	875	1	356.1	424.8	5	11
37	179		535					
.90		.95						
1021		1234						

lowest : 0 1 2 3 4, highest: 2034 2040 2095 2102 2162

d17

	n	missing	distinct	Info	Mean	Gmd	.05	.10
.25		.50	.75					
602813		0	2750	1	19.98	26	0.2500	0.5577
9618	8.1337	26.2765						1.
.90		.95						
54.0341		72.0772						

lowest : 0.000000e+00 8.733624e-03 1.265823e-02 1.869159e-02 2.040816e-02
highest: 5.079167e+02 6.607500e+02 7.510000e+02 7.570000e+02 1.267333e+03

d18

	n	missing	distinct	Info	Mean	Gmd	.05	.10
.25		.50	.75					
602813		0	205	0.936	4.892	7.405	0	0
0	1		5					
.90		.95						
13		21						

lowest : 0.0 0.5 1.0 1.5 2.0, highest: 433.0 487.0 671.0 757.0 1879.0

d19

	n	missing	distinct	Info	Mean	Gmd	.05	.10
.25		.50	.75					
602813		0	2922	1	40.38	49.33	0.7395	1.4285
5560	19.0610	59.0931						4.
.90		.95						
113.1610		139.8723						

lowest : 0.00000000 0.09304487 0.11179444 0.13543343 0.14139190
highest: 642.29259514 648.32142234 757.00000000 801.46845696 896.32000734

d20

	n	missing	distinct	Info	Mean	Gmd	.05	.10
.25		.50	.75					

```

602813      0      137      0.055      0.6379      1.269      0      0
0          0          0
.90        .95
0          0

lowest :      0      1      2      3      4, highest: 1458 1494 1841 1897 1923
-----
d21
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      509      1      432.1      505.3      5      14
43      231      683
.90        .95
1207      1398

lowest :      0      1      2      3      4, highest: 2082 2106 2129 2176 2184
-----
d22
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      2846      1      52.71      68.42      0.695      1.520      5
.146      21.677      68.197
.90        .95
141.110      191.392

lowest : 0.000000e+00 1.290323e-02 1.310044e-02 1.666667e-02 1.694915e-02
highest: 1.796250e+03 1.868500e+03 1.878333e+03 1.897000e+03 1.923000e+03
-----
d23
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      469      0.997      33.71      47.11      0      0
2          12      39
.90        .95
94          130

lowest :      0.0      0.5      1.0      1.5      2.0, highest: 1897.0 1923.0 1992.5 2
062.5 2106.0
-----
d24
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      2947      1      63.4      75.98      1.156      2.349      7
.556      31.015      96.721
.90        .95
179.450      210.952

lowest :      0.0000000      0.1128571      0.1137049      0.1280191      0.1290809
highest: 680.9620123 684.0378898 703.1440504 704.7554935 900.0684349
-----
d25

```

```

      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      800      1      -303.7      368.9      -1085      -896
-428      -139      -32
.90      .95
-9      -4

lowest : -2038 -2011 -1916 -1907 -1848, highest:      59      63      381      1311      19
23
-----
d26
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      897      1      355.1      428.9      4      10
38      164      539
.90      .95
1033      1248

lowest : -204 -101 -65 -48 -41, highest: 2102 2113 2119 2123 2131
-----
d27
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      2702      1      1.857      5.721      -3.3003      -1.3582      -0.
1446      0.2334      1.7379
.90      .95
6.1337      11.6775

lowest : -584.6667 -314.5000 -287.7500 -210.5000 -204.0000, highest: 496.60
00 639.0000 783.5000 1427.0000 1923.0000
-----
d28
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      248      0.871      -2.154      4.515      -10      -7
-2      0      0
.90      .95
0      0

lowest : -1411 -412 -290 -288 -228, highest:      531      649      783      1427      19
23
-----
d29
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      3009      1      56.58      69.44      1.036      2.108      6
.558      25.482      81.122
.90      .95
159.649      193.262

lowest : 0.000000e+00 7.516339e-02 1.290809e-01 1.561249e-01 1.855769e-01
highest: 9.665000e+02 1.086942e+03 1.224180e+03 1.240114e+03 1.386398e+03

```

```

-----
cc1
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      1841      0.995      55.86      87.15      0      0
 2      11      46
.90      .95
143      259

lowest :      0      1      2      3      4, highest: 2438 2442 2450 2459 2495
-----
cc2
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      1310      0.949      21.83      37.42      0      0
 0      2      11
.90      .95
49      105

lowest :      0      1      2      3      4, highest: 2113 2115 2119 2123 2131
-----
cc3
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      1299      0.866      19.98      35.16      0      0
 0      0      9
.90      .95
45      98

lowest :      0      1      2      3      4, highest: 2051 2076 2095 2102 2162
-----
cc4
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      1372      0.994      52.71      82.13      0      0
 2      10      44
.90      .95
136      246

lowest :      0      1      2      3      4, highest: 2123 2129 2131 2176 2184
-----
cc5
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      2397      0.987      1.857      56.72      -72      -32
-6      0      3
.90      .95
32      84

lowest : -2038 -2011 -1950 -1916 -1907, highest: 2113 2115 2119 2123 21
31

```

```

-----
basetime
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      73      1      35.46      24.21      3      7
17      35      53
.90      .95
65      69

lowest :  0  1  2  3  4, highest: 68 69 70 71 72
-----

postlength
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      1477      0.999      163.7      197.4      0      0
38      97      172
.90      .95
319      507

lowest :      0      1      2      3      4, highest: 13893 14185 14497 20016 214
80
-----

postshre
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      1639      0.988      117.3      201.1      1      1
2      13      61
.90      .95
217      450

Value      0      2000      4000      6000      8000      10000      12000      14000      18000      2
0000      26000      78000      144000
Frequency  589718      10524      1573      530      231      30      60      30      57
15      15      15      15
Proportion 0.978      0.017      0.003      0.001      0.000      0.000      0.000      0.000      0.000      0
.000      0.000      0.000      0.000
-----

postpromo
      n  missing distinct      Info      Mean      Gmd
602813      0      1      0      0      0

Value      0
Frequency  602813
Proportion 1
-----

Hhrs
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      24      0.055      23.78      0.4406      24      24
24      24      24

```

	.90	.95						
	24	24						
lowest : 1 2 3 4 5, highest: 20 21 22 23 24								

sun								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.322	73768	0.1224	0.2148	

mon								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.368	86364	0.1433	0.2455	

tue								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.382	90214	0.1497	0.2545	

wed								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.398	94825	0.1573	0.2651	

thu								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.37	86940	0.1442	0.2468	

fri								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.375	88153	0.1462	0.2497	

sat								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.355	82549	0.1369	0.2364	

basesun								
	n	missing	distinct	Info	Sum	Mean	Gmd	
602813		0	2	0.361	84427	0.1401	0.2409	

basemon								
	n	missing	distinct	Info	Sum	Mean	Gmd	


```

602813      0      2      0.348      80699      0.1339      0.2319
-----
basetue
      n  missing distinct      Info      Sum      Mean      Gmd
602813      0      2      0.356      83028      0.1377      0.2375
-----
basewed
      n  missing distinct      Info      Sum      Mean      Gmd
602813      0      2      0.376      88427      0.1467      0.2503
-----
basethu
      n  missing distinct      Info      Sum      Mean      Gmd
602813      0      2      0.392      93291      0.1548      0.2616
-----
basefri
      n  missing distinct      Info      Sum      Mean      Gmd
602813      0      2      0.371      87196      0.1446      0.2475
-----
basesat
      n  missing distinct      Info      Sum      Mean      Gmd
602813      0      2      0.366      85745      0.1422      0.244
-----
target
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
602813      0      809      0.828      7.163      12.86      0      0
0      0      3
.90      .95
12      30
lowest :      0      1      2      3      4, highest: 1702 1816 1966 2057 2106
-----
fbtest

54 Variables      1000 Observations
-----
plikes
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
1000      0      404      1      3817063      5203668      56424      184439      51
8247 1445497 4290409
.90      .95

```

```

6172812 11670473

lowest :      4233      4685      5416      6991      7394, highest: 48997484 492
03410 49408299 68687043 72606193
-----
checkin
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
1000      0      97      0.753      5355      10400      0.00      0.00
0.00    0.00    19.25
.90    .95
959.40 10826.20

lowest :      0      1      2      3      4, highest: 120918 122388
175714 185932 1100558
-----
talking
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
1000      0      402      1      269717      382284      1880      5152      2
3148      78635      277547
.90    .95
753217 1208159

lowest :      0      185      189      237      264, highest: 2454921 2891667 2
902472 3263205 6784263
-----
category
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
1000      0      74      0.974      21.88      20.27      4.00      4.00
9.00    16.00    27.00
.90    .95
46.00    61.25

lowest :      1      2      3      4      5, highest: 98 99 102 103 104
-----
d5
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
1000      0      60      0.399      8.931      17.36      0.00      0.00
0.00    0.00    0.00
.90    .95
4.00    22.05

lowest :      0      1      2      3      4, highest: 315 536 611 1218 1610
-----
d6
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75

```

```

1000      0      261      1      697.2      846.8      6.0      14.0
56.5      307.0      1128.0
.90      .95
2145.0      2369.0

lowest :      0      1      2      3      4, highest: 2730 2771 2783 2791 2858
-----
d7
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
1000      0      354      1      96.94      122.6      1.641      3.600      11
.167      53.964      118.229
.90      .95
207.505      315.710

lowest :      0.0000000      0.2500000      0.3333333      0.3636364      0.3750000
highest: 1135.5000000 1157.5000000 1218.0000000 1411.6250000 1610.0000000
-----
d8
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
1000      0      188      0.999      57.09      80.95      1.00      1.00
5.50      20.00      56.12
.90      .95
124.50      206.07

lowest :      0.0      0.5      1.0      1.5      2.0, highest: 1086.5 1101.0 1218.0 1
253.0 1610.0
-----
d9
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
1000      0      372      1      124.5      147.6      1.299      3.409      12
.211      67.124      199.594
.90      .95
303.626      369.450

lowest :      0.0000000      0.4330127      0.4714045      0.4841229      0.4948717
highest: 871.0337044 917.8454469 996.4754688 1009.5371577 1101.0000000
-----
d10
      n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
1000      0      42      0.325      4.425      8.688      0      0
0      0      0
.90      .95
1      6

lowest :      0      1      2      3      4, highest: 149 152 215 521 1610
-----
d11

```

```

      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
1000      0      250          1      580.7      736.8          4          9
38      215      937
.90   .95
1931      2160

lowest :      0      1      2      3      4, highest: 2417 2526 2655 2771 2783
-----
d12
      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
1000      0      359          1      69.2      87.53      0.9615      2.2677      8.
0000 35.7902 81.8807
.90   .95
139.6690 207.0429

lowest :      0.0000000      0.2500000      0.2857143      0.3333333      0.3636364
highest: 1000.0000000 1021.0000000 1101.0000000 1157.5000000 1610.0000000
-----
d13
      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
1000      0      150      0.999      39.58      57.93          0.0          1.0
3.0      13.0      34.0
.90   .95
83.0      141.1

lowest :      0.0      0.5      1.0      1.5      2.0, highest: 803.0 1027.0 1086.5 1
101.0 1610.0
-----
d14
      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
1000      0      370          1      99.92      120.6      1.090      2.313      9
.742 51.511 161.918
.90   .95
240.099 289.960

lowest :      0.0000000      0.4330127      0.4517540      0.4714045      0.4841229
highest: 709.3494672 803.8868080 870.1426248 881.2104402 1101.0000000
-----
d15
      n missing distinct      Info      Mean      Gmd      .05      .10
.25   .50   .75
1000      0      14      0.041      1.545      3.082          0          0
0      0      0
.90   .95
0      0


```

```

Value      0      1      2      5     10     12     23     42     43     48     56
    141    464    697
Frequency    986      2      1      1      1      1      1      1      1      1
    1      1      1
Proportion 0.986 0.002 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001 0.001
0.001 0.001 0.001
-----
d16
      n  missing distinct      Info      Mean      Gmd      .05      .10
    .25      .50      .75
    1000      0      189      0.998      277.4      394      0.00      0.00      1
1.75      64.50      340.50
    .90      .95
    837.00      1344.00

lowest :      0      1      2      3      4, highest: 1790 1810 1877 1922 2455
-----
d17
      n  missing distinct      Info      Mean      Gmd      .05      .10
    .25      .50      .75
    1000      0      287      0.998      22.8      34.07      0.000      0.000      1
.111      4.877      24.654
    .90      .95
    59.593      75.053

lowest :      0.00000000      0.06250000      0.07692308      0.11111111      0.14285714
highest: 412.37500000 420.00000000 464.00000000 471.00000000 697.00000000
-----
d18
      n  missing distinct      Info      Mean      Gmd      .05      .10
    .25      .50      .75
    1000      0      39      0.216      3.208      6.327      0.00      0.00
0.00      0.00      0.00
    .90      .95
    0.00      2.05

lowest :      0.0      0.5      1.0      1.5      2.0, highest: 186.5 341.5 353.0 464.0 697
.0
-----
d19
      n  missing distinct      Info      Mean      Gmd      .05      .10
    .25      .50      .75
    1000      0      308      0.998      49.36      69      0.000      0.000      2
.749      16.182      62.521
    .90      .95
    137.896      196.608

lowest :      0.0000000      0.2420615      0.2664694      0.3142697      0.3499271
highest: 531.7442995 580.9752442 604.2738094 619.4525809 748.9640118
-----
d20

```

```

      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      58      0.399      8.659      16.84      0      0
      0      0      0
      .90      .95
      4      22

lowest :      0      1      2      3      4, highest: 309 464 611 1162 1610
-----
d21
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      259      1      671      813.1      6.0      14.0
53.5      307.0      999.0
      .90      .95
      2040.0      2369.0

lowest :      0      1      2      3      4, highest: 2526 2608 2673 2771 2783
-----
d22
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      359      1      94.13      118.7      1.585      3.550      10
.833      52.810      118.062
      .90      .95
      198.989      311.944

lowest :      0.0000000      0.2500000      0.3333333      0.3636364      0.3750000
highest: 1101.0000000 1157.5000000 1162.0000000 1376.3750000 1610.0000000
-----
d23
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      179      0.999      55.93      79.34      1.00      1.00
5.00      19.00      55.25
      .90      .95
      118.25      206.07

lowest :      0.0      0.5      1.0      1.5      2.0, highest: 1086.5 1101.0 1162.0 1
190.5 1610.0
-----
d24
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      371      1      120      141.7      1.386      3.087      11
.473      66.076      199.259
      .90      .95
      280.460      365.209

lowest :      0.0000000      0.4330127      0.4841229      0.4948717      0.5000000
highest: 803.8868080 852.3397840 894.9410803 986.1868788 1101.0000000

```

```

-----
d25
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      180      0.999      -200.5      300.6      -890.0      -615.0      -1
83.5      -45.0      -7.0
      .90      .95
      0.0      0.0

lowest : -2119 -1782 -1677 -1422 -1348, highest:      65      79      116      136      16
10
-----

```

```

-----
d26
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      249      1      568.3      731.8      3.0      7.0
36.0      194.0      898.2
      .90      .95
      1931.0      2160.0

lowest : -392 -176 -55 -36 -35, highest: 2417 2526 2655 2771 2783
-----

```

```

-----
d27
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      348      1      46.4      70.86      -6.779      0.000      3
.278      19.750      66.000
      .90      .95
      104.483      134.168

lowest : -392.0000 -207.0000 -176.0000 -134.5000 -126.0000, highest:  961.33
33 1000.0000 1101.0000 1157.5000 1610.0000
-----

```

```

-----
d28
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      147      0.994      28.12      47.94      0      0
      1      7      22
      .90      .95
      65      99

lowest : -392.0 -176.0 -134.5 -129.0 -38.5, highest:  803.0 1027.0 1086.5 1
101.0 1610.0
-----

```

```

-----
d29
      n  missing distinct      Info      Mean      Gmd      .05      .10
      .25      .50      .75
      1000      0      374      1      116.8      137.8      1.710      3.062      11
.495      67.506      201.630
      .90      .95
      272.664      353.219
-----

```

```
lowest :    0.0000000    0.4330127    0.4517540    0.4714045    0.4841229
highest:  901.0730756  913.6934401 1101.0000000 1103.6207727 1330.2694687
```

```
-----
cc1
```

```
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
  1000      0    245    0.996    101.4    164.2      0.0      0.0
  3.0   18.0   69.0
    .90    .95
   217.1   370.1
```

```
lowest :      0      1      2      3      4, highest: 2369 2421 2635 2730 2783
```

```
-----
cc2
```

```
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
  1000      0    217    0.99    73.17    120.8      0.00      0.00
  1.00   9.50  49.25
    .90    .95
   158.20  300.00
```

```
lowest :      0      1      2      3      4, highest: 2092 2202 2311 2369 2783
```

```
-----
cc3
```

```
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
  1000      0    109    0.57    24.72    47.09      0.0      0.0
  0.0     0.0     0.0
    .90    .95
   27.1    93.3
```

```
lowest :      0      1      2      3      4, highest: 1386 1511 1583 1587 1810
```

```
-----
cc4
```

```
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
  1000      0    241    0.996    98.21    158.6      0      0
   3     17     68
    .90    .95
   217    368
```

```
lowest :      0      1      2      3      4, highest: 2202 2369 2466 2608 2783
```

```
-----
cc5
```

```
      n  missing distinct      Info      Mean      Gmd      .05      .10
.25    .50    .75
  1000      0    289    0.995    48.45    137.6     -53.1    -15.1
  0.0     5.0   39.0
    .90    .95
   136.3   257.2
```


lowest : -1296 -1047 -1018 -890 -802, highest: 1987 2092 2202 2369 2783

basetime
n missing distinct Info Mean Gmd .05 .10
.25 .50 .75
1000 0 69 0.999 17.81 18.08 0.00 1.00
4.00 13.00 28.00
.90 .95
44.00 52.05

lowest : 0 1 2 3 4, highest: 65 66 68 69 72

postlength
n missing distinct Info Mean Gmd .05 .10
.25 .50 .75
1000 0 312 1 154.7 176.3 0.0 15.0
48.0 97.5 161.0
.90 .95
265.1 390.0

lowest : 0 2 3 4 6, highest: 2399 3391 5359 5683 5858

postshre
n missing distinct Info Mean Gmd .05 .10
.25 .50 .75
1000 0 264 0.995 118.2 191.3 1.0 1.0
3.0 19.5 76.0
.90 .95
264.0 575.2

lowest : 1 2 3 4 5, highest: 2641 2962 3368 3667 3895

postpromo
n missing distinct Info Mean Gmd
1000 0 1 0 0 0

Value 0
Frequency 1000
Proportion 1

Hhrs
n missing distinct Info Mean Gmd .05 .10
.25 .50 .75
1000 0 25 0.946 14.61 9.993 1 2
6 16 24
.90 .95
24 24

lowest : 0 1 2 3 4, highest: 20 21 22 23 24

```
-----  
sun  
      n  missing distinct      Info      Sum      Mean      Gmd  
    1000      0        2    0.476      198    0.198    0.3179  
-----
```

```
-----  
mon  
      n  missing distinct      Info      Sum      Mean      Gmd  
    1000      0        2    0.2       72     0.072    0.1338  
-----
```

```
-----  
tue  
      n  missing distinct      Info      Mean      Gmd  
    1000      0        1        0        0        0  
-----
```

```
Value      0  
Frequency  1000  
Proportion 1  
-----
```

```
-----  
wed  
      n  missing distinct      Info      Mean      Gmd  
    1000      0        1        0        0        0  
-----
```

```
Value      0  
Frequency  1000  
Proportion 1  
-----
```

```
-----  
thu  
      n  missing distinct      Info      Sum      Mean      Gmd  
    1000      0        2    0.567      253    0.253    0.3784  
-----
```

```
-----  
fri  
      n  missing distinct      Info      Sum      Mean      Gmd  
    1000      0        2    0.561      249    0.249    0.3744  
-----
```

```
-----  
sat  
      n  missing distinct      Info      Sum      Mean      Gmd  
    1000      0        2    0.528      228    0.228    0.3524  
-----
```

```
-----  
basesun  
      n  missing distinct      Info      Sum      Mean      Gmd  
    1000      0        2    0.668      335    0.335    0.446  
-----  
-----
```

```

basemon
  n missing distinct      Info      Sum      Mean      Gmd
1000      0         2    0.473     196    0.196    0.3155
-----
basetue
  n missing distinct      Info      Mean      Gmd
1000      0         1         0         0         0
Value      0
Frequency 1000
Proportion 1
-----
basewed
  n missing distinct      Info      Mean      Gmd
1000      0         1         0         0         0
Value      0
Frequency 1000
Proportion 1
-----
basethu
  n missing distinct      Info      Sum      Mean      Gmd
1000      0         2    0.226      82    0.082    0.1507
-----
basefri
  n missing distinct      Info      Sum      Mean      Gmd
1000      0         2    0.449     183    0.183    0.2993
-----
basesat
  n missing distinct      Info      Sum      Mean      Gmd
1000      0         2    0.487     204    0.204    0.3251
-----
target
  n missing distinct      Info      Mean      Gmd      .05      .10
.25      .50      .75
1000      0      136    0.953    26.59    46.25    0.00    0.00
0.00      2.00    12.25
.90      .95
52.30    124.05
lowest :      0      1      2      3      4, highest: 628 741 773 1233 2136
-----
> Amelia::missmap(fbtrain); Amelia::missmap(fbtest) # no missing values
Error in loadNamespace(name) : there is no package called 'Amelia'
> train<-(fbtrain); test<-(fbtest)

```

```

> head(train); head(test)
  plikes checkin talking category d5  d6      d7 d8      d9 d10 d11
d12 d13      d14 d15 d16
1: 634995      0      463      1  0 806 11.29104  1 70.49514  0 806 7.574
627  0 69.43583  0  76
2: 634995      0      463      1  0 806 11.29104  1 70.49514  0 806 7.574
627  0 69.43583  0  76
3: 634995      0      463      1  0 806 11.29104  1 70.49514  0 806 7.574
627  0 69.43583  0  76
4: 634995      0      463      1  0 806 11.29104  1 70.49514  0 806 7.574
627  0 69.43583  0  76
5: 634995      0      463      1  0 806 11.29104  1 70.49514  0 806 7.574
627  0 69.43583  0  76
6: 634995      0      463      1  0 806 11.29104  1 70.49514  0 806 7.574
627  0 69.43583  0  76
      d17 d18      d19 d20 d21      d22 d23      d24 d25 d26      d27 d28
      d29 cc1 cc2 cc3 cc4
1: 2.604478  0 8.505502  0 806 10.64925  1 70.25479 -69 806 4.970149  0
69.85058  0  0  0  0
2: 2.604478  0 8.505502  0 806 10.64925  1 70.25479 -69 806 4.970149  0
69.85058  0  0  0  0
3: 2.604478  0 8.505502  0 806 10.64925  1 70.25479 -69 806 4.970149  0
69.85058  0  0  0  0
4: 2.604478  0 8.505502  0 806 10.64925  1 70.25479 -69 806 4.970149  0
69.85058  7  0  3  7
5: 2.604478  0 8.505502  0 806 10.64925  1 70.25479 -69 806 4.970149  0
69.85058  1  0  0  1
6: 2.604478  0 8.505502  0 806 10.64925  1 70.25479 -69 806 4.970149  0
69.85058  0  0  0  0
      cc5 basetime postlength postshre postpromo Hhrs sun mon tue wed thu fri s
at basesun basemon
1:  0      65      166      2      0  24  0  0  0  1  0  0
  0      0      0
2:  0      10      132      1      0  24  0  0  0  0  1  0
  0      0      0
3:  0      14      133      2      0  24  0  0  0  0  0  1
  0      0      0
4: -3      62      131      1      0  24  0  0  0  0  0  1
  0      0      1
5:  0      58      142      5      0  24  0  1  0  0  0  0
  0      0      0
6:  0      60      166      1      0  24  0  0  1  0  0  0
  0      0      0
      basetue basewed basethu basefri basesat target
1:  0      0      0      0      1      0
2:  0      0      0      1      0      0
3:  0      0      0      0      1      0
4:  0      0      0      0      0      0
5:  0      1      0      0      0      0
6:  0      0      0      1      0      0
      plikes checkin talking category d5  d6      d7  d8      d9 d10
d11      d12 d13
1: 1340759 33935 49392      9  0 768 77.20000 36.5 132.861432  0
 275 28.600000 14
2: 161455      0 5978      4  3  75 22.71429 15.0 23.389340  3
  75 16.857143  6

```

```

3: 92385 88 34785 70 0 0 0.00000 0.0 0.000000 0
0 0.000000 0
4: 395272 0 3406 17 0 22 4.00000 1.0 6.599663 0
22 3.111111 1
5: 4349418 0 614043 9 0 958 65.97419 24.0 126.960748 0
787 61.096774 22
6: 22899301 1 2454921 2 0 2526 958.18182 618.0 742.114157 0
2526 686.636364 565
d14 d15 d16 d17 d18 d19 d20 d21 d22 d23
d24 d25 d26
1: 45.466911 0 716 47.8250000 0 124.0505718 0 744 76.000000 36.5
129.477604 -664 275
2: 24.109444 0 15 4.1428571 0 5.4883922 3 75 21.142857 12.0
23.295791 -10 75
3: 0.000000 0 0 0.0000000 0 0.0000000 0 0 0.000000 0.0
0.000000 0 0
4: 6.740334 0 2 0.2222222 0 0.6285394 0 22 3.888889 1.0
6.573422 -2 22
5: 112.517169 0 371 4.8774194 0 33.6782375 0 912 65.587097 24.0
124.844202 -126 787
6: 694.466738 0 1583 230.0909091 0 464.4810309 0 2526 933.818182 618.0
728.681726 -1018 2526
d27 d28 d29 cc1 cc2 cc3 cc4 cc5 basetime postlength postshr
e postpromo Hhrs sun mon
1: -19.225000 5 131.036538 0 0 0 0 0 0 150 2
3 0 22 0 0
2: 12.714286 5 26.266297 22 17 5 20 12 28 20
1 0 24 0 0
3: 0.000000 0 0.000000 0 0 0 0 0 32 182
1 0 24 0 0
4: 2.888889 1 6.870944 0 0 0 0 0 30 207
1 0 20 0 0
5: 56.219355 21 107.096415 19 19 0 19 19 13 148 2
1 0 6 0 0
6: 456.545455 431 913.693440 618 618 0 618 618 7 110 18
5 0 6 0 1
tue wed thu fri sat basesun basemon basetue basewed basethu basefri bases
at target
1: 0 0 1 0 0 0 0 0 0 1 0
0 35
2: 0 0 0 1 0 0 0 0 0 0 0
1 2
3: 0 0 1 0 0 0 0 0 0 0 0
1 0
4: 0 0 0 0 1 1 0 0 0 0 0
0 0
5: 0 0 0 0 1 1 0 0 0 0 0
0 0
6: 0 0 0 0 0 0 1 0 0 0 0
0 112
> # removing overlapping observations if any
> distinct(train)
plikes checkin talking category d5 d6 d7 d8 d9 d10
d11 d12 d13
1: 634995 0 463 1 0 806 11.29104 1 70.49514 0
806 7.574627 0

```

2:	634995	0	463	1	0	806	11.29104	1	70.49514	0
806	7.574627	0								
3:	634995	0	463	1	0	806	11.29104	1	70.49514	0
806	7.574627	0								
4:	634995	0	463	1	0	806	11.29104	1	70.49514	0
806	7.574627	0								
5:	634995	0	463	1	0	806	11.29104	1	70.49514	0
806	7.574627	0								

602401:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0
1990	229.165289	79								
602402:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0
1990	229.165289	79								
602403:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0
1990	229.165289	79								
602404:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0
1990	229.165289	79								
602405:	7170111	70	497000	9	0	2164	466.97521	244	497.96617	0
1990	229.165289	79								
	d14	d15	d16	d17	d18	d19	d20	d21	d22	d23
	d24	d25	d26							
1:	69.43583	0	76	2.604478	0	8.505502	0	806	10.64925	1
70.25479	-69	806								
2:	69.43583	0	76	2.604478	0	8.505502	0	806	10.64925	1
70.25479	-69	806								
3:	69.43583	0	76	2.604478	0	8.505502	0	806	10.64925	1
70.25479	-69	806								
4:	69.43583	0	76	2.604478	0	8.505502	0	806	10.64925	1
70.25479	-69	806								
5:	69.43583	0	76	2.604478	0	8.505502	0	806	10.64925	1
70.25479	-69	806								

602401:	386.58908	0	1734	174.595041	60	298.763750	0	2082	446.80165	240
472.06568	-1348	1990								
602402:	386.58908	0	1734	174.595041	60	298.763750	0	2082	446.80165	240
472.06568	-1348	1990								
602403:	386.58908	0	1734	174.595041	60	298.763750	0	2082	446.80165	240
472.06568	-1348	1990								
602404:	386.58908	0	1734	174.595041	60	298.763750	0	2082	446.80165	240
472.06568	-1348	1990								
602405:	386.58908	0	1734	174.595041	60	298.763750	0	2082	446.80165	240
472.06568	-1348	1990								
	d27	d28	d29	cc1	cc2	cc3	cc4	cc5	basetime	postlength
postshre	postpromo	Hhrs								
1:	4.970149	0	69.85058	0	0	0	0	0	65	166
2		0	24							
2:	4.970149	0	69.85058	0	0	0	0	0	10	132
1		0	24							
3:	4.970149	0	69.85058	0	0	0	0	0	14	133
2		0	24							
4:	4.970149	0	69.85058	7	0	3	7	-3	62	131
1		0	24							
5:	4.970149	0	69.85058	1	0	0	1	0	58	142
5		0	24							

```

---
602401: 54.570248 -29 506.79360 1940 1940 0 1940 1940 16 77
2123 0 24
602402: 54.570248 -29 506.79360 155 155 0 155 155 4 130
474 0 24
602403: 54.570248 -29 506.79360 207 207 0 207 207 12 12
1511 0 24
602404: 54.570248 -29 506.79360 629 85 544 619 -459 31 149
1099 0 24
602405: 54.570248 -29 506.79360 691 77 614 652 -537 42 120
2962 0 10
sun mon tue wed thu fri sat basesun basemon basetue basewed basethu
basefri basesat target
1: 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0
2: 0 0 0 0 1 0 0 0 0 0 0 0
1 0 0
3: 0 0 0 0 0 1 0 0 0 0 0 0
0 1 0
4: 0 0 0 0 0 1 0 0 1 0 0 0
0 0 0
5: 0 1 0 0 0 0 0 0 0 0 1 0
0 0 0
---
602401: 0 0 1 0 0 0 0 0 0 0 1 0
0 0 203
602402: 0 0 1 0 0 0 0 0 0 0 1 0
0 0 42
602403: 0 0 0 1 0 0 0 0 0 0 1 0
0 0 51
602404: 0 0 0 1 0 0 0 0 0 0 0 1
0 0 17
602405: 0 0 0 1 0 0 0 0 0 0 0 0
1 0 21
> dim(train)
[1] 602813 54
> distinct(test)
plikes checkin talking category d5 d6 d7 d8 d9 d1
0 d11 d12 d13
1: 1340759 33935 49392 9 0 768 77.20000 36.5 132.861432
0 275 28.600000 14.0
2: 161455 0 5978 4 3 75 22.71429 15.0 23.389340
3 75 16.857143 6.0
3: 92385 88 34785 70 0 0 0.00000 0.0 0.000000
0 0 0.000000 0.0
4: 395272 0 3406 17 0 22 4.00000 1.0 6.599663
0 22 3.111111 1.0
5: 4349418 0 614043 9 0 958 65.97419 24.0 126.960748
0 787 61.096774 22.0
---
996: 220578 0 26719 92 0 53 11.16667 5.5 14.079141
0 40 5.791667 2.0

```

```

997: 160261      0      4347      28  1  173  28.75000    7.0  54.876566
0   9   2.625000  2.0
998: 4253181     40  472926      9 23 1470 274.78571 103.0 378.943678  2
3 1470 207.857143 93.5
999: 952756      8  443107      9  0 2369  83.03738   18.0 238.545669
0 2369  81.046729 17.5
1000: 4716837    0  490422      5  0 1128  81.87850   30.0 164.593450
0   616  53.299065 24.0
      d14 d15 d16      d17 d18      d19 d20 d21      d22 d2
3      d24 d25 d26
1:  45.466911    0 716 47.8250000 0.0 124.0505718    0 744 76.000000 36.
5 129.477604 -664 275
2:  24.109444    0 15  4.1428571 0.0   5.4883922    3  75 21.142857 12.
0 23.295791 -10   75
3:   0.000000    0  0  0.0000000 0.0   0.0000000    0  0  0.000000  0.
0  0.000000    0  0
4:   6.740334    0  2  0.2222222 0.0   0.6285394    0 22  3.888889  1.
0  6.573422  -2   22
5: 112.517169    0 371 4.8774194 0.0  33.6782375    0 912 65.587097 24.
0 124.844202 -126 787
---

996: 10.923900    0 26  4.2916667 0.0   7.0974125    0 49 10.833333  5.
5 13.535960 -22   40
997:  2.642797    0 170 23.8750000 1.5  55.4040556    1 173 28.250000  7.
0 55.015339 -167   9
998: 360.293503    0 611 56.9285714 0.0 158.4065584  23 1470 270.357143 103.
0 377.372610 -473 1470
999: 237.071961    0 218  1.9906542 0.0  17.3284625    0 2369  82.990654 18.
0 238.472906 -163 2369
1000: 87.861532    0 837 25.8317757 0.0 114.1808277    0 1083  79.177570 30.
0 155.558674 -615 616
      d27 d28      d29 cc1 cc2 cc3 cc4 cc5 basetime postlength pos
tshre postpromo Hhrs sun
1: -19.225000    5.0 131.036538    0  0  0  0  0      0      150
23      0  22   0
2: 12.714286    5.0 26.266297  22 17  5 20 12      28      20
1      0  24   0
3:  0.000000    0.0  0.000000    0  0  0  0  0      32      182
1      0  24   0
4:  2.888889    1.0  6.870944    0  0  0  0  0      30      207
1      0  20   0
5: 56.219355   21.0 107.096415  19 19  0 19 19      13      148
21      0   6   0
---

996:  1.500000    0.0 14.074208  53  4 26 49 -22      51      69
16      0  24   0
997: -21.250000  -1.0 55.339294   1  1  0  1  1      3      13
13      0  20   1
998: 150.928571 76.5 407.129054  63 63  0 63 63      17      177
20      0   4   0
999: 79.056075   17.0 236.860179   2  2  0  2  2      1      122
1      0  14   1
1000: 27.467290  10.0 122.844200   5  5  0  5  5      8      108
38      0  24   0

```



```

      mon tue wed thu fri sat basesun basemon basetue basewed basethu basefr
i basesat target
1:  0  0  0  1  0  0      0      0      0      0      1
0      0      35
2:  0  0  0  0  1  0      0      0      0      0      0
0      1      2
3:  0  0  0  1  0  0      0      0      0      0      0
0      1      0
4:  0  0  0  0  0  1      1      0      0      0      0
0      0      0
5:  0  0  0  0  0  1      1      0      0      0      0
0      0      0
---
996:  0  0  0  1  0  0      0      0      0      0      0
0      1      0
997:  0  0  0  0  0  0      1      0      0      0      0
0      0      1
998:  0  0  0  1  0  0      0      0      0      0      0
1      0      0
999:  0  0  0  0  0  0      1      0      0      0      0
0      0      0
1000:  0  0  0  1  0  0      0      0      0      0      1
0      0      5
> dim(test)
[1] 1000  54

```

list the levels for the class

```
sapply(train, class)
```

missing values count

```
sapply(train, function(x) sum(is.na(x))) # no missing values
```

model building

```
library(lars)
```

```
x<-as.matrix(train[,c(30:34, 36:38)])
```

```
y<-as.matrix(train[,35])
```

```
fittrain<-lars(x,y,type = "lasso")
```

```
summary(fittrain)
```

```
plot(fittrain)
```

```

> distinct(test)
      plikes checkin talking category d5  d6      d7  d8      d9 d1
0  d11      d12  d13

```

```

1: 1340759 33935 49392 9 0 768 77.20000 36.5 132.861432
0 275 28.600000 14.0
2: 161455 0 5978 4 3 75 22.71429 15.0 23.389340
3 75 16.857143 6.0
3: 92385 88 34785 70 0 0 0.00000 0.0 0.000000
0 0 0.000000 0.0
4: 395272 0 3406 17 0 22 4.00000 1.0 6.599663
0 22 3.111111 1.0
5: 4349418 0 614043 9 0 958 65.97419 24.0 126.960748
0 787 61.096774 22.0
---

996: 220578 0 26719 92 0 53 11.16667 5.5 14.079141
0 40 5.791667 2.0
997: 160261 0 4347 28 1 173 28.75000 7.0 54.876566
0 9 2.625000 2.0
998: 4253181 40 472926 9 23 1470 274.78571 103.0 378.943678 2
3 1470 207.857143 93.5
999: 952756 8 443107 9 0 2369 83.03738 18.0 238.545669
0 2369 81.046729 17.5
1000: 4716837 0 490422 5 0 1128 81.87850 30.0 164.593450
0 616 53.299065 24.0
d14 d15 d16 d17 d18 d19 d20 d21 d22 d2
3 d24 d25 d26
1: 45.466911 0 716 47.8250000 0.0 124.0505718 0 744 76.000000 36.
5 129.477604 -664 275
2: 24.109444 0 15 4.1428571 0.0 5.4883922 3 75 21.142857 12.
0 23.295791 -10 75
3: 0.000000 0 0 0.0000000 0.0 0.0000000 0 0 0.000000 0.
0 0.000000 0 0
4: 6.740334 0 2 0.2222222 0.0 0.6285394 0 22 3.888889 1.
0 6.573422 -2 22
5: 112.517169 0 371 4.8774194 0.0 33.6782375 0 912 65.587097 24.
0 124.844202 -126 787
---

996: 10.923900 0 26 4.2916667 0.0 7.0974125 0 49 10.833333 5.
5 13.535960 -22 40
997: 2.642797 0 170 23.8750000 1.5 55.4040556 1 173 28.250000 7.
0 55.015339 -167 9
998: 360.293503 0 611 56.9285714 0.0 158.4065584 23 1470 270.357143 103.
0 377.372610 -473 1470
999: 237.071961 0 218 1.9906542 0.0 17.3284625 0 2369 82.990654 18.
0 238.472906 -163 2369
1000: 87.861532 0 837 25.8317757 0.0 114.1808277 0 1083 79.177570 30.
0 155.558674 -615 616
d27 d28 d29 cc1 cc2 cc3 cc4 cc5 basetime postlength pos
tshre postpromo Hhrs sun
1: -19.225000 5.0 131.036538 0 0 0 0 0 0 150
23 0 22 0
2: 12.714286 5.0 26.266297 22 17 5 20 12 28 20
1 0 24 0
3: 0.000000 0.0 0.000000 0 0 0 0 0 32 182
1 0 24 0
4: 2.888889 1.0 6.870944 0 0 0 0 0 30 207
1 0 20 0

```

```

5: 56.219355 21.0 107.096415 19 19 0 19 19 13 148
21 0 6 0
---

996: 1.500000 0.0 14.074208 53 4 26 49 -22 51 69
16 0 24 0
997: -21.250000 -1.0 55.339294 1 1 0 1 1 3 13
13 0 20 1
998: 150.928571 76.5 407.129054 63 63 0 63 63 17 177
20 0 4 0
999: 79.056075 17.0 236.860179 2 2 0 2 2 1 122
1 0 14 1
1000: 27.467290 10.0 122.844200 5 5 0 5 5 8 108
38 0 24 0

mon tue wed thu fri sat basesun basemon basetue basewed basethu basefr
i basesat target
1: 0 0 0 1 0 0 0 0 0 0 0 1
0 0 35
2: 0 0 0 0 1 0 0 0 0 0 0 0
0 1 2
3: 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0
4: 0 0 0 0 0 1 1 0 0 0 0 0
0 0 0
5: 0 0 0 0 0 1 1 0 0 0 0 0
0 0 0
---

996: 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0
997: 0 0 0 0 0 0 1 0 0 0 0 0
0 0 1
998: 0 0 0 1 0 0 0 0 0 0 0 0
1 0 0
999: 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0
1000: 0 0 0 1 0 0 0 0 0 0 0 1
0 0 5

> dim(test)
[1] 1000 54
> # list the levels for the class
> sapply(train, class)
 plikes checkin talking category d5 d6 d7
 d8 d9
"integer" "integer" "integer" "integer" "numeric" "numeric" "numeric"
"numeric" "numeric"
 d10 d11 d12 d13 d14 d15 d16
 d17 d18
"numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
"numeric" "numeric"
 d19 d20 d21 d22 d23 d24 d25
 d26 d27
"numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
"numeric" "numeric"
 d28 d29 cc1 cc2 cc3 cc4 cc5
 basetime postlength

```

```

"numeric" "numeric" "integer" "integer" "integer" "integer" "integer"
"integer" "integer"
postshre postpromo      Hhrs      sun      mon      tue      wed
thu      fri
"integer" "integer" "integer" "integer" "integer" "integer" "integer"
"integer" "integer"
sat      basesun      basemon      basetue      basewed      basethu      basefri
basesat      target
"integer" "integer" "integer" "integer" "integer" "integer" "integer"
"integer" "integer"
> # missing values count
> sapply(train, function(x) sum(is.na(x))) # no missing values
      plikes      checkin      talking      category      d5      d6      d7
      d8      d9
      0      0      0      0      0      0      0
      0      0
      d10      d11      d12      d13      d14      d15      d16
      d17      d18
      0      0      0      0      0      0      0
      0      0
      d19      d20      d21      d22      d23      d24      d25
      d26      d27
      0      0      0      0      0      0      0
      0      0
      d28      d29      cc1      cc2      cc3      cc4      cc5
      basetime postlength
      0      0      0      0      0      0      0
      0      0
      postshre postpromo      Hhrs      sun      mon      tue      wed
      thu      fri
      0      0      0      0      0      0      0
      0      0
      sat      basesun      basemon      basetue      basewed      basethu      basefri
      basesat      target
      0      0      0      0      0      0      0
      0      0

> # model building
> library(lars)
> x<-as.matrix(train[,c(30:34, 36:38)])
> y<-as.matrix(train[,35])
> fittrain<-lars(x,y,type = "lasso")
> summary(fittrain)
LARS/LASSO
Call: lars(x = x, y = y, type = "lasso")
      Df      Rss      Cp
0  1 265184072 86443.704
1  2 257888480 67483.326
2  3 250899583 49320.092
3  4 241002262 23597.415
4  3 237154106 13593.457
5  4 232167426 634.301
6  5 231951415 74.853
7  6 231946914 65.154
8  7 231923385 6.000
9  8 231923385 8.000
> plot(fittrain)

```

```
Error in plot.new() : figure margins too large
```

```
#result fit
```

```
fittrain
```

```
fittrain
Call:
lars(x = x, y = y, type = "lasso")
R-squared: 0.125
Sequence of LASSO moves:
      cc5 cc2 cc1 cc5 cc3 postshre postlength cc4 cc5
Var      5  2  1 -5  3      7      6  4  5
Step     1  2  3  4  5      6      7  8  9
```

```
# select a step with a minimum error
```

```
best_step <- fittrain$df[which.min(fittrain$RSS)]
```

```
best_step
```

```
summary(best_step)
```

```
> # select a step with a minimum error
> best_step <- fittrain$df[which.min(fittrain$RSS)]
> best_step

7
summary(best_step)
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      7      7      7      7      7      7
```

```
# predictions
```

```
predictions <- predict(fittrain, x, s=best_step, type="fit")$fittrain
```

```
# summarize accuracy
```

```
mse_train <- mean((y - predictions)^2)
```

```
print(mse_train)
```

```
# Elastic Net
```

```
install.packages("elasticnet")
```

```
library(elasticnet)
```

```
elasticnet::
```

```
  ENreg.fit(x_train,y_train)
```

```
pred_cv = ENreg.predict(x_cv)
```

```
#calculating mse
```

```
mse = np.mean((pred_cv - y_cv)*2)
```

```
#mse 1773750.73
```

```
ENreg.score(x_cv,y_cv)
```

```
# leaner model
```

```
LM<-lm(target~ x+y, data = train)
```

```
summary(LM)
```

```
attributes(LM)
```

```
plot(LM)
```

```
coef(LM)
```

```
> print(mse_train)
[1] NaN
> # Elastic Net
> install.packages("elasticnet")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Jagannath/Documents/R/win-library/3.6'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/elasticnet_1.1.1.zip'
Content type 'application/zip' length 234980 bytes (229 KB)
downloaded 229 KB

package 'elasticnet' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
      C:\Users\Jagannath\AppData\Local\Temp\RtmpwnSTYR\downloaded_packages
> library(elasticnet)
> elasticnet::
Error: unexpected end of line in "elasticnet::"
> ENreg.fit(x_train,y_train)
Error in ENreg.fit(x_train, y_train) :
  could not find function "ENreg.fit"
> pred_cv = ENreg.predict(x_cv)
Error in ENreg.predict(x_cv) : could not find function "ENreg.predict"
> mse = np.mean((pred_cv - y_cv)*2)
Error in np.mean((pred_cv - y_cv) * 2) :
  could not find function "np.mean"
> ENreg.score(x_cv,y_cv)
Error in ENreg.score(x_cv, y_cv) : could not find function "ENreg.score"
```

```

> # leaner model
> LM<-lm(target~ x+y, data = train)
> summary(LM)

Call:
lm(formula = target ~ x + y, data = train)

Residuals:
    Min       1Q   Median       3Q      Max
-468.83   -5.43    -1.19     2.22  2096.33

Coefficients: (2 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.872e+00  7.924e-02  111.959  <2e-16 ***
xcca1         9.310e-02  3.655e-03   25.474  <2e-16 ***
xcca2         2.326e-01  9.076e-04  256.266  <2e-16 ***
xcca3        -2.164e-02  8.694e-04  -24.884  <2e-16 ***
xcca4        -9.299e-02  4.068e-03  -22.862  <2e-16 ***
xcca5                NA          NA      NA      NA
xpostlength   7.273e-05  9.847e-05    0.739    0.46
xpostshre     1.806e-03  3.994e-05   45.220  <2e-16 ***
xpostpromo                NA          NA      NA      NA
y             -1.940e-01  1.886e-03 -102.821  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.73 on 602805 degrees of freedom
Multiple R-squared:  0.3051, Adjusted R-squared:  0.3051
F-statistic: 3.78e+04 on 7 and 602805 DF, p-value: < 2.2e-16

> attributes(LM)
$names
[1] "coefficients" "residuals"      "effects"        "rank"          "fitted
.values" "assign"
[7] "qr"           "df.residual"    "xlevels"        "call"          "terms"
      "model"

$class
[1] "lm"

> plot(LM)
Error in plot.new() : figure margins too large
> coef(LM)
      (Intercept)      xcca1      xcca2      xcca3      xcca4
      xcca5 xpostlength
8.872030e+00  9.310113e-02  2.325740e-01 -2.163527e-02 -9.299280e-02
      NA  7.273452e-05
      xpostshre xpostpromo      y
1.806077e-03      NA -1.939668e-01
> plot(train$target, data=train, main="Scatterplot")

```

```
plot(train$target, data=train, main="Scatterplot")
```

```

Loading required package: Matrix
> input<- weight_lifting_exercises
> View(input)
> input1<- as.numeric(input$new_window)
Warning message:
NAs introduced by coercion
> model<- glm(input1~raw_timestamp_part_1+raw_timestamp_part_2+cvtd_timestamp+num_window+roll_belt+pitch_belt+yaw_belt+total_accel_belt,data = input)
Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) :
  contrasts can be applied only to factors with 2 or more levels
> model
Error: object 'model' not found
> summary(model)
Error in summary(model) : object 'model' not found
> predict<- predict(model, type = "response")
Error in predict(model, type = "response") : object 'model' not found
> head(predict, 5)

1 function (object, ...)
2 UseMethod("predict")
> input$predict<- predict
Error in rep(value, length.out = nrows) :
  attempt to replicate an object of type 'closure'
> input$predictROUND<- round(predict, digits = 0)
Error in round(predict, digits = 0) :
  non-numeric argument to mathematical function
> table(input$new_window, predict>= 0.5)
Error in predict >= 0.5 :
  comparison (5) is possible only for atomic and list types
> dim(input)
[1] 4024    59
> #1. Use the below given data set
> #Data Set
> library(readr)
> library(data.table)
data.table 1.12.2 using 4 threads (see ?getDTthreads). Latest news: r-datatable.com
Warning message:
package 'data.table' was built under R version 3.6.1
> library(tidyr)

Attaching package: 'tidyr'

The following object is masked from 'package:Matrix':

    expand

> install.packages("tidyverse")
WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/Jagannath/Documents/R/win-library/3.6'
(as 'lib' is unspecified)

```



```

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/tidyverse_1.2.1.zip'
Content type 'application/zip' length 92983 bytes (90 KB)
downloaded 90 KB

package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\Jagannath\AppData\Local\Temp\RtmpwnSTYR\downloaded_packages
> library(tidyverse)
-- Attaching packages ----- tidyverse 1.2.1 --
v ggplot2 3.2.0      v dplyr 0.8.3
v tibble 2.1.3       v stringr 1.4.0
v purrr 0.3.2        v forcats 0.4.0
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::between()   masks data.table::between()
x tidyr::expand()    masks Matrix::expand()
x dplyr::filter()    masks stats::filter()
x dplyr::first()     masks data.table::first()
x dplyr::lag()       masks stats::lag()
x dplyr::last()      masks data.table::last()
x purrr::transpose() masks data.table::transpose()
Warning messages:
1: package 'tidyverse' was built under R version 3.6.1
2: package 'dplyr' was built under R version 3.6.1
3: package 'forcats' was built under R version 3.6.1
> library(caret)
Loading required package: lattice

Attaching package: 'caret'

The following object is masked from 'package:purrr':

  lift

Warning message:
package 'caret' was built under R version 3.6.1
> library(glmnet)
Loading required package: foreach

Attaching package: 'foreach'

The following objects are masked from 'package:purrr':

  accumulate, when

Loaded glmnet 2.0-18

Warning messages:
1: package 'glmnet' was built under R version 3.6.1
2: package 'foreach' was built under R version 3.6.1
> install.packages("mlbench")

```

WARNING: Rtools is required to build R packages but is not currently installed. Please download and install the appropriate version of Rtools before proceeding:

```
https://cran.rstudio.com/bin/windows/Rtools/  
Installing package into 'C:/Users/Jagannath/Documents/R/win-library/3.6'  
(as 'lib' is unspecified)  
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.6/mlbench_2.1-1.zip'  
Content type 'application/zip' length 1061202 bytes (1.0 MB)  
downloaded 1.0 MB
```

package 'mlbench' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\Jagannath\AppData\Local\Temp\RtmpwnSTYR\downloaded_packages

```
> library(mlbench)
```

Warning message:

package 'mlbench' was built under R version 3.6.1

```
> getwd()
```

```
[1] "C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/  
assignment 16-20"
```

```
> data1<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_1.csv")
```

```
> data2<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_2.csv")
```

```
> data3<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_3.csv")
```

```
> data4<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_4.csv")
```

```
> data5<- fread("C:/Users/Jagannath/Documents/assignment data acadgild/assignment 16-20/assignment 16-20/task 16 data/data task16/Training/Features_Variant_5.csv")
```

```
> train_set<-rbind(data1,data2,data3,data4,data5)
```

```
> train_set1<-na.omit(train_set)
```

```
> train_set1
```

		V1	V2		V3	V4	V5	V6		V7	V8		V9	V10	V11
V12	V13		V14	V15											
	1:	634995	0		463	1	0	806	11.29104	1	70.49514	0	806	7.57	
4627	0	69.43583	0												
	2:	634995	0		463	1	0	806	11.29104	1	70.49514	0	806	7.57	
4627	0	69.43583	0												
	3:	634995	0		463	1	0	806	11.29104	1	70.49514	0	806	7.57	
4627	0	69.43583	0												
	4:	634995	0		463	1	0	806	11.29104	1	70.49514	0	806	7.57	
4627	0	69.43583	0												
	5:	634995	0		463	1	0	806	11.29104	1	70.49514	0	806	7.57	
4627	0	69.43583	0												

```
602809: 7170111 70 497000 9 0 2164 466.97521 244 497.96617 0 1990 229.16  
5289 79 386.58908 0
```

```

602810: 7170111 70 497000 9 0 2164 466.97521 244 497.96617 0 1990 229.16
5289 79 386.58908 0
602811: 7170111 70 497000 9 0 2164 466.97521 244 497.96617 0 1990 229.16
5289 79 386.58908 0
602812: 7170111 70 497000 9 0 2164 466.97521 244 497.96617 0 1990 229.16
5289 79 386.58908 0
602813: 7170111 70 497000 9 0 2164 466.97521 244 497.96617 0 1990 229.16
5289 79 386.58908 0
      V16      V17 V18      V19 V20  V21      V22 V23      V24  V2
5  V26      V27 V28
  1: 76 2.604478 0 8.505502 0 806 10.64925 1 70.25479 -6
9 806 4.970149 0
  2: 76 2.604478 0 8.505502 0 806 10.64925 1 70.25479 -6
9 806 4.970149 0
  3: 76 2.604478 0 8.505502 0 806 10.64925 1 70.25479 -6
9 806 4.970149 0
  4: 76 2.604478 0 8.505502 0 806 10.64925 1 70.25479 -6
9 806 4.970149 0
  5: 76 2.604478 0 8.505502 0 806 10.64925 1 70.25479 -6
9 806 4.970149 0
    ---

602809: 1734 174.595041 60 298.763750 0 2082 446.80165 240 472.06568 -134
8 1990 54.570248 -29
602810: 1734 174.595041 60 298.763750 0 2082 446.80165 240 472.06568 -134
8 1990 54.570248 -29
602811: 1734 174.595041 60 298.763750 0 2082 446.80165 240 472.06568 -134
8 1990 54.570248 -29
602812: 1734 174.595041 60 298.763750 0 2082 446.80165 240 472.06568 -134
8 1990 54.570248 -29
602813: 1734 174.595041 60 298.763750 0 2082 446.80165 240 472.06568 -134
8 1990 54.570248 -29
      V29 V30 V31 V32 V33 V34 V35 V36 V37 V38 V39 V40 V41 V42 V
43 V44 V45 V46 V47 V48
  1: 69.85058 0 0 0 0 0 65 166 2 0 24 0 0 0
1 0 0 0 0 0
  2: 69.85058 0 0 0 0 0 10 132 1 0 24 0 0 0
0 1 0 0 0 0
  3: 69.85058 0 0 0 0 0 14 133 2 0 24 0 0 0
0 0 1 0 0 0
  4: 69.85058 7 0 3 7 -3 62 131 1 0 24 0 0 0
0 0 1 0 0 1
  5: 69.85058 1 0 0 1 0 58 142 5 0 24 0 1 0
0 0 0 0 0 0
    ---

602809: 506.79360 1940 1940 0 1940 1940 16 77 2123 0 24 0 0 1
0 0 0 0 0 0
602810: 506.79360 155 155 0 155 155 4 130 474 0 24 0 0 1
0 0 0 0 0 0
602811: 506.79360 207 207 0 207 207 12 12 1511 0 24 0 0 0
1 0 0 0 0 0
602812: 506.79360 629 85 544 619 -459 31 149 1099 0 24 0 0 0
1 0 0 0 0 0
602813: 506.79360 691 77 614 652 -537 42 120 2962 0 10 0 0 0
1 0 0 0 0 0

```

	V49	V50	V51	V52	V53	V54
1:	0	0	0	0	1	0
2:	0	0	0	1	0	0
3:	0	0	0	0	1	0
4:	0	0	0	0	0	0
5:	0	1	0	0	0	0

602809:	0	1	0	0	0	203
602810:	0	1	0	0	0	42
602811:	0	1	0	0	0	51
602812:	0	0	1	0	0	17
602813:	0	0	0	1	0	21

Task 2:

1. Use the below-given data set

Data Set

2. Perform the below-given activities:
 - a. Create a classification model using a logistic regression model
 - b. verify model goodness of fit
 - c. Report the accuracy measures
 - d. Report the variable importance
 - e. Report the unimportant variables
 - f. Interpret the results
 - g. Visualize the results

```
# assignment 16-20
```

```
#task 2
```

```
weight_lifting_exercises<- weight_lifting
View('weight_lifting_exercises')
input<- weight_lifting_exercises
View(input)
input1<- as.numeric(input$new_window)
model<-
glm(input1~raw_timestamp_part_1+raw_timestamp_part_2+cvtd_timestamp+num_window+roll_belt+pitch_belt+yaw_belt+total_accel_belt,data = input)
model
summary(model)
predict<- predict(model, type = "response")
head(predict, 5)
input$predict<- predict
input$predictROUND<- round(predict, digits = 0)
table(input$new_window, predict>= 0.5)
dim(input)
predict<- predict(model, type = "response")
```

```
head(predict, 5)
```

```
##### above is from nirlipta's file .. below is some one from github
```

```
#-----Assignment 17 -----
```

```
# Weight Lifting Exercise Analysis
```

```
# Import Data Set
```

```
data_set <-
```

```
read.csv("Example_WearableComputing_weight_lifting_exercises_biceps_curl_variations  
task2.csv")
```

```
View(data_set)
```

```
# remove irrelevant columns viz. name, cvtd_timestamp, new_window
```

```
data <- weight_lifting
```

```
View(data)
```

```
str(data)
```

```
# 2. Perform the below given activities:
```

```
# a. Create classification model using logistic regression model
```

```
# the target variable variable is multiple level
```

```
sum(is.na(data)) # there are no missing values
```

```
# splitting the data set for train and test
```

```
library(caTools)
```

```
set.seed(123)
```

```
split = sample.split(data$classe, SplitRatio = 0.7)
```

```
train = subset(data, split == TRUE)      # train data
```

```
test = subset(data, split == FALSE)      # test data
```

```
library(nnet) ; library(MASS)
```

```
model <- multinom(classe ~., data = train)
```

```
summary(model)
```

```
# stepAIC(model, direction = "backward")
```

```
final <- multinom(classe ~ raw_timestamp_part_1 + num_window + roll_belt + pitch_belt +  
yaw_belt + total_accel_belt + gyros_belt_x + gyros_belt_y +  
gyros_belt_z + accel_belt_x + accel_belt_y + accel_belt_z +  
magnet_belt_x + magnet_belt_y + magnet_belt_z + roll_arm +
```

```

pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x + gyros_arm_y +
gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z + magnet_arm_y +
magnet_arm_z + roll_dumbbell + pitch_dumbbell + yaw_dumbbell +
gyros_dumbbell_x + gyros_dumbbell_z + accel_dumbbell_x +
accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm + pitch_forearm +
yaw_forearm + total_accel_forearm + gyros_forearm_x + gyros_forearm_y +
gyros_forearm_z + accel_forearm_x + accel_forearm_y + accel_forearm_z +
magnet_forearm_x + magnet_forearm_y + magnet_forearm_z, data = train)

final
summary(final)

# Predictions
predicted <- predict(final, newdata= test)

# -----

# b. Goodness of Fit
library(car)
chisq.test(table(test$classe), prop.table(table(predicted)))

```

```

> model<- glm(input1~raw_timestamp_part_1+raw_timestamp_part_2+cvtd_timestam
p+num_window+roll_belt+pitch_belt+yaw_belt+total_accel_belt,data = input)
Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) :
  contrasts can be applied only to factors with 2 or more levels
> model
> summary(model)
> predict<- predict(model, type = "response")
> head(predict, 5)

1 function (object, ...)
2 UseMethod("predict")
> input$predict<- predict
> input$predictROUND<- round(predict, digits = 0)
> table(input$new_window, predict>= 0.5)
> dim(input)
[1] 4024 159
> predict<- predict(model, type = "response")
Error in predict(model, type = "response") : object 'model' not found
> head(predict, 5)

1 function (object, ...)
2 UseMethod("predict")
> data_set <- read.csv("Example_WearableComputing_weight_lifting_exercises_b
iceps_curl_variations task2.csv")
> View(data_set)
> # remove irrelevant collumns viz. name, cvtd_timestamp, new_window
> data <- weight_lifting
> View(data)

```

```

> str(data)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    4024 obs. of  1
59 variables:
 $ user_name           : chr  "eurico" "eurico" "eurico" "eurico" ...
 $ raw_timestamp_part_1 : num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+
09 ...
 $ raw_timestamp_part_2 : num  34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp       : chr  "28/11/2011 14:15" "28/11/2011 14:15" "28/
11/2011 14:15" "28/11/2011 14:15" ...
 $ new_window           : chr  "no" "no" "no" "no" ...
 $ num_window           : num  1 1 1 1 1 1 1 1 ...
 $ roll_belt            : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.3
1 2 ...
 $ pitch_belt           : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47
.4 47.7 ...
 $ yaw_belt             : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9
-82.2 -82.6 -82.8 ...
 $ total_accel_belt     : num  3 2 1 1 1 1 3 4 2 3 ...
 $ kurtosis_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ kurtosis_pitch_belt  : chr  NA NA NA NA ...
 $ kurtosis_yaw_belt    : chr  NA NA NA NA ...
 $ skewness_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ skewness_roll_belt_1 : chr  NA NA NA NA ...
 $ skewness_yaw_belt    : chr  NA NA NA NA ...
 $ max_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ max_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ max_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ min_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ min_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ min_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_yaw_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA ...
 $ avg_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_roll_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ var_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ avg_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_pitch_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
 $ var_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ avg_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_yaw_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
 $ var_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ gyros_belt_x         : num  2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.6
5 1.48 ...
 $ gyros_belt_y         : num  0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0
.03 -0.06 ...
 $ gyros_belt_z         : num  0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.
33 0.21 ...
 $ accel_belt_x         : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
 $ accel_belt_y         : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
 $ accel_belt_z         : num  22 16 8 7 0 -5 -8 -9 -7 1 ...
 $ magnet_belt_x        : num  387 405 409 422 418 432 438 440 443 449 ..
.
 $ magnet_belt_y        : num  525 512 511 513 508 510 508 503 507 499 ..
.

```



```

$ magnet_belt_z      : num  -267 -254 -244 -221 -208 -189 -176 -163 -1
40 -132 ...
$ roll_arm           : num   132 129 125 120 115 110 104 98.6 93.2 88.5
...
$ pitch_arm          : num   -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9
-49.7 -49 -48.1 ...
$ yaw_arm            : num   -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -1
1.4 -4.49 1.82 ...
$ total_accel_arm    : num    38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm    : num    NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm   : num    NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm     : num    NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x        : num    2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4
3.48 ...
$ gyros_arm_y        : num   -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83
-0.83 -0.83 -0.8 ...
$ gyros_arm_z        : num   -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -
0.21 -0.11 -0.15 ...
$ accel_arm_x        : num   143 146 156 158 163 160 165 153 143 135 ..
.
$ accel_arm_y        : num    30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z        : num   -346 -339 -307 -305 -288 -274 -225 -218 -2
05 -134 ...
$ magnet_arm_x       : num   556 599 613 646 670 696 721 725 740 741 ..
.
$ magnet_arm_y       : num   -205 -206 -198 -186 -175 -174 -161 -152 -1
33 -115 ...
$ magnet_arm_z       : num   -374 -335 -319 -268 -241 -193 -121 -105 -4
3 14 ...
$ kurtosis_roll_arm  : chr    NA NA NA NA ...
$ kurtosis_pitch_arm : chr    NA NA NA NA ...
$ kurtosis_yaw_arm   : chr    NA NA NA NA ...
$ skewness_roll_arm  : chr    NA NA NA NA ...
$ skewness_pitch_arm : chr    NA NA NA NA ...
$ skewness_yaw_arm   : chr    NA NA NA NA ...
$ max_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm : num    NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm : num    NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm  : num    NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell      : num    51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell     : num    11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell       : num    104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num    NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_dumbbell : num    NA NA NA NA NA NA NA NA NA NA ...

```

```

$ kurtosis_yaw_dumbbell : chr NA NA NA NA ...
$ skewness_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell : chr NA NA NA NA ...
$ max_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]
- attr(*, "spec")=
.. cols(
..   user_name = col_character(),
..   raw_timestamp_part_1 = col_double(),
..   raw_timestamp_part_2 = col_double(),
..   cvtd_timestamp = col_character(),
..   new_window = col_character(),
..   num_window = col_double(),
..   roll_belt = col_double(),
..   pitch_belt = col_double(),
..   yaw_belt = col_double(),
..   total_accel_belt = col_double(),
..   kurtosis_roll_belt = col_double(),
..   kurtosis_pitch_belt = col_character(),
..   kurtosis_yaw_belt = col_character(),
..   skewness_roll_belt = col_double(),
..   skewness_roll_belt_1 = col_character(),
..   skewness_yaw_belt = col_character(),
..   max_roll_belt = col_double(),
..   max_pitch_belt = col_double(),
..   max_yaw_belt = col_double(),
..   min_roll_belt = col_double(),
..   min_pitch_belt = col_double(),
..   min_yaw_belt = col_double(),
..   amplitude_roll_belt = col_double(),
..   amplitude_pitch_belt = col_double(),
..   amplitude_yaw_belt = col_double(),
..   var_total_accel_belt = col_double(),
..   avg_roll_belt = col_double(),
..   stddev_roll_belt = col_double(),
..   var_roll_belt = col_double(),
..   avg_pitch_belt = col_double(),
..   stddev_pitch_belt = col_double(),
..   var_pitch_belt = col_double(),
..   avg_yaw_belt = col_double(),
..   stddev_yaw_belt = col_double(),
..   var_yaw_belt = col_double(),
..   gyros_belt_x = col_double(),
..   gyros_belt_y = col_double(),
..   gyros_belt_z = col_double(),
..   accel_belt_x = col_double(),
..   accel_belt_y = col_double(),
..   accel_belt_z = col_double(),

```

```
.. magnet_belt_x = col_double(),
.. magnet_belt_y = col_double(),
.. magnet_belt_z = col_double(),
.. roll_arm = col_double(),
.. pitch_arm = col_double(),
.. yaw_arm = col_double(),
.. total_accel_arm = col_double(),
.. var_accel_arm = col_double(),
.. avg_roll_arm = col_double(),
.. stddev_roll_arm = col_double(),
.. var_roll_arm = col_double(),
.. avg_pitch_arm = col_double(),
.. stddev_pitch_arm = col_double(),
.. var_pitch_arm = col_double(),
.. avg_yaw_arm = col_double(),
.. stddev_yaw_arm = col_double(),
.. var_yaw_arm = col_double(),
.. gyros_arm_x = col_double(),
.. gyros_arm_y = col_double(),
.. gyros_arm_z = col_double(),
.. accel_arm_x = col_double(),
.. accel_arm_y = col_double(),
.. accel_arm_z = col_double(),
.. magnet_arm_x = col_double(),
.. magnet_arm_y = col_double(),
.. magnet_arm_z = col_double(),
.. kurtosis_roll_arm = col_character(),
.. kurtosis_pitch_arm = col_character(),
.. kurtosis_yaw_arm = col_character(),
.. skewness_roll_arm = col_character(),
.. skewness_pitch_arm = col_character(),
.. skewness_yaw_arm = col_character(),
.. max_roll_arm = col_double(),
.. max_pitch_arm = col_double(),
.. max_yaw_arm = col_double(),
.. min_roll_arm = col_double(),
.. min_pitch_arm = col_double(),
.. min_yaw_arm = col_double(),
.. amplitude_roll_arm = col_double(),
.. amplitude_pitch_arm = col_double(),
.. amplitude_yaw_arm = col_double(),
.. roll_dumbbell = col_double(),
.. pitch_dumbbell = col_double(),
.. yaw_dumbbell = col_double(),
.. kurtosis_roll_dumbbell = col_double(),
.. kurtosis_pitch_dumbbell = col_double(),
.. kurtosis_yaw_dumbbell = col_character(),
.. skewness_roll_dumbbell = col_double(),
.. skewness_pitch_dumbbell = col_double(),
.. skewness_yaw_dumbbell = col_character(),
.. max_roll_dumbbell = col_double(),
.. max_pitch_dumbbell = col_double(),
.. max_yaw_dumbbell = col_double(),
.. min_roll_dumbbell = col_double(),
.. min_pitch_dumbbell = col_double(),
.. min_yaw_dumbbell = col_double(),
```

```
.. amplitude_roll_dumbbell = col_double(),
.. amplitude_pitch_dumbbell = col_double(),
.. amplitude_yaw_dumbbell = col_double(),
.. total_accel_dumbbell = col_double(),
.. var_accel_dumbbell = col_double(),
.. avg_roll_dumbbell = col_double(),
.. stddev_roll_dumbbell = col_double(),
.. var_roll_dumbbell = col_double(),
.. avg_pitch_dumbbell = col_double(),
.. stddev_pitch_dumbbell = col_double(),
.. var_pitch_dumbbell = col_double(),
.. avg_yaw_dumbbell = col_double(),
.. stddev_yaw_dumbbell = col_double(),
.. var_yaw_dumbbell = col_double(),
.. gyros_dumbbell_x = col_double(),
.. gyros_dumbbell_y = col_double(),
.. gyros_dumbbell_z = col_double(),
.. accel_dumbbell_x = col_double(),
.. accel_dumbbell_y = col_double(),
.. accel_dumbbell_z = col_double(),
.. magnet_dumbbell_x = col_double(),
.. magnet_dumbbell_y = col_double(),
.. magnet_dumbbell_z = col_double(),
.. roll_forearm = col_double(),
.. pitch_forearm = col_double(),
.. yaw_forearm = col_double(),
.. kurtosis_roll_forearm = col_character(),
.. kurtosis_pitch_forearm = col_character(),
.. kurtosis_yaw_forearm = col_character(),
.. skewness_roll_forearm = col_character(),
.. skewness_pitch_forearm = col_character(),
.. skewness_yaw_forearm = col_character(),
.. max_roll_forearm = col_double(),
.. max_pitch_forearm = col_double(),
.. max_yaw_forearm = col_double(),
.. min_roll_forearm = col_double(),
.. min_pitch_forearm = col_double(),
.. min_yaw_forearm = col_double(),
.. amplitude_roll_forearm = col_double(),
.. amplitude_pitch_forearm = col_double(),
.. amplitude_yaw_forearm = col_double(),
.. total_accel_forearm = col_double(),
.. var_accel_forearm = col_double(),
.. avg_roll_forearm = col_double(),
.. stddev_roll_forearm = col_double(),
.. var_roll_forearm = col_double(),
.. avg_pitch_forearm = col_double(),
.. stddev_pitch_forearm = col_double(),
.. var_pitch_forearm = col_double(),
.. avg_yaw_forearm = col_double(),
.. stddev_yaw_forearm = col_double(),
.. var_yaw_forearm = col_double(),
.. gyros_forearm_x = col_double(),
.. gyros_forearm_y = col_double(),
.. gyros_forearm_z = col_double(),
.. accel_forearm_x = col_double(),
```

```

.. accel_forearm_y = col_double(),
.. accel_forearm_z = col_double(),
.. magnet_forearm_x = col_double(),
.. magnet_forearm_y = col_double(),
.. magnet_forearm_z = col_double(),
.. classe = col_character()
.. )
> sum(is.na(data)) # there are no missing values
[1] 393600
> library(caTools)
Warning message:
package 'caTools' was built under R version 3.6.1
> set.seed(123)
> split = sample.split(data$classe, SplitRatio = 0.7)
> train = subset(data, split == TRUE) # train data
> test = subset(data, split == FALSE) # test data
> library(nnet) ; library(MASS)
> model <- multinom(classe ~., data = train)
Error in `contrasts<-`(`*tmp*`, value = contr.funs[1 + isOF[nn]]) :
  contrasts can be applied only to factors with 2 or more levels
> summary(model)
Error in summary(model) : object 'model' not found
> final <- multinom(classe ~ raw_timestamp_part_1 + num_window + roll_belt +
  pitch_belt +
+ yaw_belt + total_accel_belt + gyros_belt_x + gyros_belt_y +
+ gyros_belt_z + accel_belt_x + accel_belt_y + accel_belt_z +
+ magnet_belt_x + magnet_belt_y + magnet_belt_z + roll_arm +
+ pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x +
+ gyros_arm_y + gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z +
+ magnet_arm_x + magnet_arm_y + magnet_arm_z + roll_dumbbell + pitch_dumbbell + yaw_dumbbell +
+ gyros_dumbbell_x + gyros_dumbbell_y + gyros_dumbbell_z + accel_dumbbell_x +
+ accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
+ magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm +
+ pitch_forearm + yaw_forearm + total_accel_forearm + gyros_forearm_x +
+ gyros_forearm_y + gyros_forearm_z + accel_forearm_x + accel_forearm_y +
+ accel_forearm_z + magnet_forearm_x + magnet_forearm_y + magnet_forearm_z
, data = train)
# weights: 265 (208 variable)
initial value 4532.177161
iter 10 value 1695.817687
iter 20 value 823.145181
iter 30 value 724.071332
iter 40 value 357.227728
iter 50 value 289.194690
iter 50 value 289.194690

```

```

iter 60 value 201.646078
iter 70 value 105.292085
iter 80 value 76.164403
iter 90 value 57.973059
iter 100 value 49.430197
final value 49.430197
stopped after 100 iterations
> final
Call:
multinom(formula = classe ~ raw_timestamp_part_1 + num_window +
  roll_belt + pitch_belt + yaw_belt + total_accel_belt + gyros_belt_x +
  gyros_belt_y + gyros_belt_z + accel_belt_x + accel_belt_y +
  accel_belt_z + magnet_belt_x + magnet_belt_y + magnet_belt_z +
  roll_arm + pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x +
  gyros_arm_y + gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z +
  magnet_arm_y + magnet_arm_z + roll_dumbbell + pitch_dumbbell +
  yaw_dumbbell + gyros_dumbbell_x + gyros_dumbbell_z + accel_dumbbell_x +
  accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
  magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm + pitch_forearm +
  yaw_forearm + total_accel_forearm + gyros_forearm_x + gyros_forearm_y +
  gyros_forearm_z + accel_forearm_x + accel_forearm_y + accel_forearm_z +
  magnet_forearm_x + magnet_forearm_y + magnet_forearm_z, data = train)

Coefficients:
      (Intercept) raw_timestamp_part_1  num_window      roll_belt  pitch_belt
yaw_belt
B  2.236996e-09      -1.363158e-09  0.09105180 -0.0048339198 -0.001141864
-0.005088154
C  4.494416e-09      -2.195099e-08  0.04043045  0.0060653121 -0.001833431
0.010231355
D -4.286441e-10      -2.797434e-08 -0.02413039 -0.0008663083  0.005287776
-0.006181941
E  3.340052e-08      2.144261e-08  0.02269082 -0.0036644520 -0.006990474
0.009706589
  total_accel_belt  gyros_belt_x  gyros_belt_y  gyros_belt_z  accel_belt_x  ac
cel_belt_y  accel_belt_z
B   -0.0034695300 -1.725407e-04  3.202847e-05 -1.791988e-04  0.005522384  0
.004046689  0.019442560
C    0.0011722424  2.711960e-04 -1.643575e-05  2.464993e-05 -0.002186109  0
.002454132 -0.008387581
D   -0.0015899760  9.948438e-05 -6.790312e-05 -5.471208e-04 -0.006243753  0
.003994443  0.010726061
E    0.0009475916 -5.727850e-05  7.624242e-05  6.062901e-04  0.011859843 -0
.008821349 -0.002486073
  magnet_belt_x  magnet_belt_y  magnet_belt_z      roll_arm  pitch_arm  ya
w_arm  total_accel_arm
B  -0.021459955  0.0280579103  0.071911618  0.003798006 -0.06576813 -0.0250
97131  -0.0208380039
C   0.007047188 -0.0007373633 -0.008946121  0.019143671  0.03876987  0.0239
03727  -0.0117818993
D  -0.004494439  0.0014234456 -0.001797440 -0.002427299 -0.01689625 -0.0017
80834  0.0003814782
E   0.025647016 -0.0056357674  0.016470815  0.019009746  0.01069149  0.0230
81535  -0.0010059266
  gyros_arm_x  gyros_arm_y  gyros_arm_z  accel_arm_x  accel_arm_y  accel_
arm_z  magnet_arm_y

```

```

B -0.0136631649 7.976717e-03 0.0001970857 -0.05704138 0.021008371 0.0434
76960 0.031921773
C 0.0025245032 -2.212564e-03 0.0002045377 -0.01685926 -0.004174773 0.0078
57859 -0.051624345
D 0.0004537182 1.180763e-05 -0.0002996302 0.03288859 -0.001635021 -0.0165
28067 -0.018772245
E 0.0008257342 -6.560845e-04 0.0002976939 -0.03988844 0.022885775 0.0372
78992 -0.002718483
magnet_arm_z roll_dumbbell pitch_dumbbell yaw_dumbbell gyros_dumbbell_x gy
ros_dumbbell_z
B -0.06237601 -0.007141826 -0.020049687 0.008375728 -0.0008487920
0.0016483160
C 0.01507292 -0.004422265 0.004605364 0.005811493 0.0003230252
-0.0012762521
D 0.02225466 0.023570751 0.010568385 0.027368428 -0.0001990213
-0.0003307281
E -0.03772053 -0.030077984 -0.017389277 -0.021913405 0.0002969471
0.0001542316
accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magne
t_dumbbell_y
B 0.004025183 0.01887781 0.01304531 0.006073832
-0.024380679
C -0.011994461 -0.03696102 0.03148393 -0.011762556
-0.020482381
D -0.006274208 -0.02829597 -0.01092769 -0.022455773
0.048481796
E 0.035197146 0.04121934 0.03266806 -0.019823353
-0.005658537
magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm total_accel_fore
arm gyros_forearm_x
B -0.004437042 0.023174767 0.0004713158 -0.016731752 0.020793
016 2.453234e-03
C -0.010555326 0.041341274 0.0003490628 -0.004882358 0.012333
997 6.541527e-05
D -0.018808938 -0.003670976 0.0138553269 0.018909946 0.007584
437 1.026190e-04
E 0.036497263 0.007166679 0.0013250337 0.016361177 -0.012458
468 -4.443229e-04
gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y accel_fore
arm_z magnet_forearm_x
B -1.096523e-02 -1.697679e-03 0.004155049 0.03401914 0.0041
83016 0.015509573
C -5.693345e-05 -9.110254e-05 0.029864708 0.02577970 0.0025
81164 0.017710500
D 7.816097e-04 1.800418e-04 -0.040467072 -0.01966517 0.0081
47917 -0.010237055
E -1.034282e-04 7.203492e-05 0.019704081 0.02519823 0.0212
96631 -0.003299594
magnet_forearm_y magnet_forearm_z
B -0.03992453 0.0463339909
C -0.04187582 0.0488198263
D -0.01242610 0.0110447932
E -0.01985137 -0.0003666157

Residual Deviance: 98.86039
AIC: 514.8604

```

```

> summary(final)
Call:
multinom(formula = classe ~ raw_timestamp_part_1 + num_window +
  roll_belt + pitch_belt + yaw_belt + total_accel_belt + gyros_belt_x +
  gyros_belt_y + gyros_belt_z + accel_belt_x + accel_belt_y +
  accel_belt_z + magnet_belt_x + magnet_belt_y + magnet_belt_z +
  roll_arm + pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x +
  gyros_arm_y + gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z +
  magnet_arm_y + magnet_arm_z + roll_dumbbell + pitch_dumbbell +
  yaw_dumbbell + gyros_dumbbell_x + gyros_dumbbell_z + accel_dumbbell_x +
  accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
  magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm + pitch_forearm +
  yaw_forearm + total_accel_forearm + gyros_forearm_x + gyros_forearm_y +
  gyros_forearm_z + accel_forearm_x + accel_forearm_y + accel_forearm_z +
  magnet_forearm_x + magnet_forearm_y + magnet_forearm_z, data = train)

Coefficients:
  (Intercept) raw_timestamp_part_1  num_window      roll_belt  pitch_belt
yaw_belt
B  2.236996e-09      -1.363158e-09  0.09105180 -0.0048339198 -0.001141864
-0.005088154
C  4.494416e-09      -2.195099e-08  0.04043045  0.0060653121 -0.001833431
0.010231355
D -4.286441e-10      -2.797434e-08 -0.02413039 -0.0008663083  0.005287776
-0.006181941
E  3.340052e-08      2.144261e-08  0.02269082 -0.0036644520 -0.006990474
0.009706589
  total_accel_belt  gyros_belt_x  gyros_belt_y  gyros_belt_z  accel_belt_x  ac
cel_belt_y  accel_belt_z
B  -0.0034695300 -1.725407e-04  3.202847e-05 -1.791988e-04  0.005522384  0
.004046689  0.019442560
C   0.0011722424  2.711960e-04 -1.643575e-05  2.464993e-05 -0.002186109  0
.002454132 -0.008387581
D  -0.0015899760  9.948438e-05 -6.790312e-05 -5.471208e-04 -0.006243753  0
.003994443  0.010726061
E   0.0009475916 -5.727850e-05  7.624242e-05  6.062901e-04  0.011859843 -0
.008821349 -0.002486073
  magnet_belt_x  magnet_belt_y  magnet_belt_z      roll_arm  pitch_arm  ya
w_arm  total_accel_arm
B  -0.021459955  0.0280579103  0.071911618  0.003798006 -0.06576813 -0.0250
97131  -0.0208380039
C   0.007047188 -0.0007373633 -0.008946121  0.019143671  0.03876987  0.0239
03727  -0.0117818993
D  -0.004494439  0.0014234456 -0.001797440 -0.002427299 -0.01689625 -0.0017
80834  0.0003814782
E   0.025647016 -0.0056357674  0.016470815  0.019009746  0.01069149  0.0230
81535  -0.0010059266
  gyros_arm_x  gyros_arm_y  gyros_arm_z  accel_arm_x  accel_arm_y  accel_
arm_z  magnet_arm_y
B -0.0136631649  7.976717e-03  0.0001970857 -0.05704138  0.021008371  0.0434
76960  0.031921773
C  0.0025245032 -2.212564e-03  0.0002045377 -0.01685926 -0.004174773  0.0078
57859 -0.051624345
D  0.0004537182  1.180763e-05 -0.0002996302  0.03288859 -0.001635021 -0.0165
28067 -0.018772245

```



```

E 0.0008257342 -6.560845e-04 0.0002976939 -0.03988844 0.022885775 0.0372
78992 -0.002718483
magnet_arm_z roll_dumbbell pitch_dumbbell yaw_dumbbell gyros_dumbbell_x gy
ros_dumbbell_z
B -0.06237601 -0.007141826 -0.020049687 0.008375728 -0.0008487920
0.0016483160
C 0.01507292 -0.004422265 0.004605364 0.005811493 0.0003230252
-0.0012762521
D 0.02225466 0.023570751 0.010568385 0.027368428 -0.0001990213
-0.0003307281
E -0.03772053 -0.030077984 -0.017389277 -0.021913405 0.0002969471
0.0001542316
accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magne
t_dumbbell_y
B 0.004025183 0.01887781 0.01304531 0.006073832
-0.024380679
C -0.011994461 -0.03696102 0.03148393 -0.011762556
-0.020482381
D -0.006274208 -0.02829597 -0.01092769 -0.022455773
0.048481796
E 0.035197146 0.04121934 0.03266806 -0.019823353
-0.005658537
magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm total_accel_fore
arm gyros_forearm_x
B -0.004437042 0.023174767 0.0004713158 -0.016731752 0.020793
016 2.453234e-03
C -0.010555326 0.041341274 0.0003490628 -0.004882358 0.012333
997 6.541527e-05
D -0.018808938 -0.003670976 0.0138553269 0.018909946 0.007584
437 1.026190e-04
E 0.036497263 0.007166679 0.0013250337 0.016361177 -0.012458
468 -4.443229e-04
gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y accel_fore
arm_z magnet_forearm_x
B -1.096523e-02 -1.697679e-03 0.004155049 0.03401914 0.0041
83016 0.015509573
C -5.693345e-05 -9.110254e-05 0.029864708 0.02577970 0.0025
81164 0.017710500
D 7.816097e-04 1.800418e-04 -0.040467072 -0.01966517 0.0081
47917 -0.010237055
E -1.034282e-04 7.203492e-05 0.019704081 0.02519823 0.0212
96631 -0.003299594
magnet_forearm_y magnet_forearm_z
B -0.03992453 0.0463339909
C -0.04187582 0.0488198263
D -0.01242610 0.0110447932
E -0.01985137 -0.0003666157

Std. Errors:
(Intercept) raw_timestamp_part_1 num_window roll_belt pitch_belt
yaw_belt
B 1.289777e-19 1.706501e-10 9.503846e-18 1.627086e-17 3.507766e-18 1
.004031e-18
C 2.457477e-19 3.251476e-10 2.141019e-17 3.013940e-17 6.333887e-18 7
.733794e-19

```

```

D 1.215895e-18      1.608732e-09 3.226374e-17 8.186366e-19 7.674979e-18 1
.133883e-16
E 1.176744e-18      1.556938e-09 5.258754e-17 7.251690e-17 1.773989e-17 5
.968324e-17
  total_accel_belt gyros_belt_x gyros_belt_y gyros_belt_z accel_belt_x accel
_belt_y accel_belt_z
B 2.590896e-18 5.886290e-20 4.532352e-21 5.873163e-20 5.471791e-18 8.906
424e-18 2.323839e-17
C 4.798937e-18 1.162626e-19 7.216148e-21 1.115351e-19 1.016891e-17 1.743
652e-17 4.253783e-17
D 3.587810e-18 1.105971e-20 1.850568e-20 1.345357e-19 2.103331e-17 1.482
614e-18 2.389211e-17
E 1.291258e-17 2.916276e-19 1.565030e-20 7.216347e-19 3.459927e-17 4.549
685e-17 9.785106e-17
  magnet_belt_x magnet_belt_y magnet_belt_z      roll_arm      pitch_arm      y
aw_arm total_accel_arm
B 3.022570e-18 7.393457e-17 4.997686e-17 5.546128e-17 6.704723e-18 8.4309
89e-17 2.831770e-18
C 3.491811e-19 1.436349e-16 9.066805e-17 2.717741e-17 3.391661e-18 1.2187
72e-17 6.081187e-18
D 5.165961e-19 7.294147e-16 3.779746e-16 7.955097e-17 3.819338e-17 4.6379
00e-17 1.974668e-17
E 1.292492e-17 6.783128e-16 4.534427e-16 3.481909e-17 2.546126e-17 8.3621
40e-17 3.405447e-17
  gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_ar
m_z magnet_arm_y
B 9.098726e-19 4.398910e-19 3.790563e-20 3.406499e-17 2.437789e-17 4.092270e
-17 6.069455e-17
C 8.190068e-20 1.283379e-19 2.117975e-20 3.889764e-17 3.617581e-18 7.966507e
-18 5.111921e-17
D 1.701341e-18 5.079584e-19 3.522974e-19 1.510558e-17 1.056285e-16 4.273232e
-17 6.369871e-19
E 1.100189e-18 6.683111e-19 6.916372e-19 6.585145e-17 1.910074e-17 1.414372e
-16 2.086086e-16
  magnet_arm_z roll_dumbbell pitch_dumbbell yaw_dumbbell gyros_dumbbell_x gy
ros_dumbbell_z
B 8.232175e-17 1.070253e-17 9.512748e-18 2.115266e-17 6.521144e-20
4.000531e-20
C 6.588564e-17 1.423756e-17 2.307518e-18 1.984333e-17 1.384238e-19
1.034230e-19
D 4.151328e-16 4.319632e-17 1.132785e-17 1.148090e-16 6.019352e-20
2.340041e-20
E 3.387306e-16 7.416724e-17 7.384508e-17 6.434542e-17 8.611212e-20
6.020422e-21
  accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magne
t_dumbbell_y
B 9.857717e-18 1.119280e-17 1.209581e-17 8.781590e-17
6.346295e-17
C 4.338462e-18 1.524084e-17 1.590679e-17 8.537480e-17
1.520607e-16
D 8.378879e-18 2.333674e-17 2.476604e-17 6.271992e-16
3.956658e-16
E 1.350387e-16 1.005679e-16 3.640172e-17 4.319807e-16
1.237012e-16
  magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm total_accel_fore
arm gyros_forearm_x

```

```

B      1.610612e-17 2.202720e-17 1.103112e-17 1.380901e-17      4.246450e
-18    2.668999e-19
C      3.576154e-17 1.602152e-17 4.297630e-18 1.188324e-17      9.409439e
-18    1.855843e-20
D      5.877386e-18 8.511953e-17 4.837392e-17 7.485575e-17      4.345010e
-17    2.933953e-20
E      6.799727e-17 1.107671e-16 5.238683e-17 1.906894e-17      4.302260e
-17    3.149971e-19
      gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y accel_fore
arm_z magnet_forearm_x
B      1.328172e-18 2.902488e-19      5.022358e-17      7.569670e-17      2.19505
7e-17      8.864237e-17
C      1.844195e-19 8.866300e-20      1.547405e-17      4.174947e-17      4.42071
2e-17      6.386250e-17
D      1.016503e-18 3.094208e-19      1.434339e-16      2.539318e-16      2.24280
3e-16      6.590090e-16
E      1.951326e-18 5.880521e-19      2.938107e-16      2.678555e-16      2.66217
2e-16      8.307565e-17
      magnet_forearm_y magnet_forearm_z
B      2.788504e-16      1.339348e-16
C      9.695325e-17      1.430405e-16
D      3.833278e-16      6.270757e-16
E      9.354359e-16      5.448831e-16

Residual Deviance: 98.86039
AIC: 514.8604
> # Predictions
> predicted <- predict(final, newdata= test)

```

```

> library(car)
> chisq.test(table(test$classe), prop.table(table(predicted)))

      Pearson's Chi-squared test

data:  table(test$classe) and prop.table(table(predicted))
X-squared = 20, df = 16, p-value = 0.2202

```

```

# -----
# c. Report the accuracy measures
# Accuracy
conf <- table(test$classe, predicted)
OAA <- (conf[1,1]+conf[2,2]+conf[3,3]+conf[4,4]+conf[5,5]) / sum(conf)
OAA

# -----
# d. Report the variable importance

```

```

coef(final)
library(caret)
varImp(final)

# -----
# e. Report the unimportant variables
install.packages("Rtools")
install_github("riv", "tomasgreif")
install_github("woe", "tomasgreif")

library(devtools); library(woe); library(riv)

iv_df <- iv.mult(train, y = "classe", summary= FALSE, verbose = TRUE)

variables <- c(colnames(train[, -56]))
imp_variables <- names(as.data.frame(coef(final)))

unimportant_variables <- setdiff(variables, imp_variables)
unimportant_variables

```

```

> final <- multinom(classe ~ raw_timestamp_part_1 + num_window + roll_belt +
  pitch_belt +
  + yaw_belt + total_accel_belt + gyros_belt_x + gyros_belt_y +
  t_y + gyros_belt_z + accel_belt_x + accel_belt_y + accel_belt_z +
  t_z + magnet_belt_x + magnet_belt_y + magnet_belt_z + roll_arm +
  + pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x +
  gyros_arm_y + gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z +
  + magnet_arm_x + magnet_arm_y + magnet_arm_z + roll_dumbbell + pitch_dumbbell + yaw_dumbbell +
  mbbell + gyros_dumbbell_x + gyros_dumbbell_z + accel_dumbbell_x +
  + accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
  x + magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm +
  pitch_forearm + yaw_forearm + total_accel_forearm + gyros_forearm_x +
  + gyros_forearm_y + gyros_forearm_z + accel_forearm_x + accel_forearm_y +
  accel_forearm_z + magnet_forearm_x + magnet_forearm_y + magnet_forearm_z
  , data = train)
# weights: 265 (208 variable)
initial value 4532.177161
iter 10 value 1695.817687

```

```

iter 20 value 823.145181
iter 30 value 724.071332
iter 40 value 357.227728
iter 50 value 289.194690
iter 50 value 289.194690
iter 60 value 201.646078
iter 70 value 105.292085
iter 80 value 76.164403
iter 90 value 57.973059
iter 100 value 49.430197
final value 49.430197
stopped after 100 iterations
> final
Call:
multinom(formula = classe ~ raw_timestamp_part_1 + num_window +
  roll_belt + pitch_belt + yaw_belt + total_accel_belt + gyros_belt_x +
  gyros_belt_y + gyros_belt_z + accel_belt_x + accel_belt_y +
  accel_belt_z + magnet_belt_x + magnet_belt_y + magnet_belt_z +
  roll_arm + pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x +
  gyros_arm_y + gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z +
  magnet_arm_y + magnet_arm_z + roll_dumbbell + pitch_dumbbell +
  yaw_dumbbell + gyros_dumbbell_x + gyros_dumbbell_z + accel_dumbbell_x +
  accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
  magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm + pitch_forearm +
  yaw_forearm + total_accel_forearm + gyros_forearm_x + gyros_forearm_y +
  gyros_forearm_z + accel_forearm_x + accel_forearm_y + accel_forearm_z +
  magnet_forearm_x + magnet_forearm_y + magnet_forearm_z, data = train)

Coefficients:
      (Intercept) raw_timestamp_part_1  num_window      roll_belt  pitch_belt
yaw_belt
B  2.236996e-09      -1.363158e-09  0.09105180 -0.0048339198 -0.001141864
-0.005088154
C  4.494416e-09      -2.195099e-08  0.04043045  0.0060653121 -0.001833431
0.010231355
D -4.286441e-10      -2.797434e-08 -0.02413039 -0.0008663083  0.005287776
-0.006181941
E  3.340052e-08       2.144261e-08  0.02269082 -0.0036644520 -0.006990474
0.009706589
  total_accel_belt  gyros_belt_x  gyros_belt_y  gyros_belt_z  accel_belt_x  ac
cel_belt_y  accel_belt_z
B   -0.0034695300 -1.725407e-04  3.202847e-05 -1.791988e-04  0.005522384  0
.004046689  0.019442560
C    0.0011722424  2.711960e-04 -1.643575e-05  2.464993e-05 -0.002186109  0
.002454132 -0.008387581
D   -0.0015899760  9.948438e-05 -6.790312e-05 -5.471208e-04 -0.006243753  0
.003994443  0.010726061
E    0.0009475916 -5.727850e-05  7.624242e-05  6.062901e-04  0.011859843 -0
.008821349 -0.002486073
  magnet_belt_x  magnet_belt_y  magnet_belt_z      roll_arm  pitch_arm  ya
w_arm  total_accel_arm
B  -0.021459955  0.0280579103  0.071911618  0.003798006 -0.06576813 -0.0250
97131  -0.0208380039
C   0.007047188 -0.0007373633 -0.008946121  0.019143671  0.03876987  0.0239
03727  -0.0117818993

```

D	-0.004494439	0.0014234456	-0.001797440	-0.002427299	-0.01689625	-0.0017
80834	0.0003814782					
E	0.025647016	-0.0056357674	0.016470815	0.019009746	0.01069149	0.0230
81535	-0.0010059266					
	gyros_arm_x	gyros_arm_y	gyros_arm_z	accel_arm_x	accel_arm_y	accel_arm_z
	magnet_arm_y					
B	-0.0136631649	7.976717e-03	0.0001970857	-0.05704138	0.021008371	0.0434
76960	0.031921773					
C	0.0025245032	-2.212564e-03	0.0002045377	-0.01685926	-0.004174773	0.0078
57859	-0.051624345					
D	0.0004537182	1.180763e-05	-0.0002996302	0.03288859	-0.001635021	-0.0165
28067	-0.018772245					
E	0.0008257342	-6.560845e-04	0.0002976939	-0.03988844	0.022885775	0.0372
78992	-0.002718483					
	magnet_arm_z	roll_dumbbell	pitch_dumbbell	yaw_dumbbell	gyros_dumbbell_x	gyros_dumbbell_z
B	-0.06237601	-0.007141826	-0.020049687	0.008375728	-0.0008487920	
	0.0016483160					
C	0.01507292	-0.004422265	0.004605364	0.005811493	0.0003230252	
	-0.0012762521					
D	0.02225466	0.023570751	0.010568385	0.027368428	-0.0001990213	
	-0.0003307281					
E	-0.03772053	-0.030077984	-0.017389277	-0.021913405	0.0002969471	
	0.0001542316					
	accel_dumbbell_x	accel_dumbbell_y	accel_dumbbell_z	magnet_dumbbell_x	magnet_dumbbell_y	
B	0.004025183	0.01887781	0.01304531	0.006073832		
	-0.024380679					
C	-0.011994461	-0.03696102	0.03148393	-0.011762556		
	-0.020482381					
D	-0.006274208	-0.02829597	-0.01092769	-0.022455773		
	0.048481796					
E	0.035197146	0.04121934	0.03266806	-0.019823353		
	-0.005658537					
	magnet_dumbbell_z	roll_forearm	pitch_forearm	yaw_forearm	total_accel_forearm	
	gyros_forearm_x					
B	-0.004437042	0.023174767	0.0004713158	-0.016731752	0.020793	
016	2.453234e-03					
C	-0.010555326	0.041341274	0.0003490628	-0.004882358	0.012333	
997	6.541527e-05					
D	-0.018808938	-0.003670976	0.0138553269	0.018909946	0.007584	
437	1.026190e-04					
E	0.036497263	0.007166679	0.0013250337	0.016361177	-0.012458	
468	-4.443229e-04					
	gyros_forearm_y	gyros_forearm_z	accel_forearm_x	accel_forearm_y	accel_forearm_z	
	magnet_forearm_x					
B	-1.096523e-02	-1.697679e-03	0.004155049	0.03401914	0.0041	
83016	0.015509573					
C	-5.693345e-05	-9.110254e-05	0.029864708	0.02577970	0.0025	
81164	0.017710500					
D	7.816097e-04	1.800418e-04	-0.040467072	-0.01966517	0.0081	
47917	-0.010237055					
E	-1.034282e-04	7.203492e-05	0.019704081	0.02519823	0.0212	
96631	-0.003299594					
	magnet_forearm_y	magnet_forearm_z				
B	-0.03992453	0.0463339909				

```
C      -0.04187582      0.0488198263
D      -0.01242610      0.0110447932
E      -0.01985137     -0.0003666157
```

Residual Deviance: 98.86039

AIC: 514.8604

```
> summary(final)
```

Call:

```
multinom(formula = classe ~ raw_timestamp_part_1 + num_window +
  roll_belt + pitch_belt + yaw_belt + total_accel_belt + gyros_belt_x +
  gyros_belt_y + gyros_belt_z + accel_belt_x + accel_belt_y +
  accel_belt_z + magnet_belt_x + magnet_belt_y + magnet_belt_z +
  roll_arm + pitch_arm + yaw_arm + total_accel_arm + gyros_arm_x +
  gyros_arm_y + gyros_arm_z + accel_arm_x + accel_arm_y + accel_arm_z +
  magnet_arm_y + magnet_arm_z + roll_dumbbell + pitch_dumbbell +
  yaw_dumbbell + gyros_dumbbell_x + gyros_dumbbell_z + accel_dumbbell_x +
  accel_dumbbell_y + accel_dumbbell_z + magnet_dumbbell_x +
  magnet_dumbbell_y + magnet_dumbbell_z + roll_forearm + pitch_forearm +
  yaw_forearm + total_accel_forearm + gyros_forearm_x + gyros_forearm_y +
  gyros_forearm_z + accel_forearm_x + accel_forearm_y + accel_forearm_z +
  magnet_forearm_x + magnet_forearm_y + magnet_forearm_z, data = train)
```

Coefficients:

```
(Intercept) raw_timestamp_part_1 num_window roll_belt pitch_belt
yaw_belt
B 2.236996e-09 -1.363158e-09 0.09105180 -0.0048339198 -0.001141864
-0.005088154
C 4.494416e-09 -2.195099e-08 0.04043045 0.0060653121 -0.001833431
0.010231355
D -4.286441e-10 -2.797434e-08 -0.02413039 -0.0008663083 0.005287776
-0.006181941
E 3.340052e-08 2.144261e-08 0.02269082 -0.0036644520 -0.006990474
0.009706589
total_accel_belt gyros_belt_x gyros_belt_y gyros_belt_z accel_belt_x ac
cel_belt_y accel_belt_z
B -0.0034695300 -1.725407e-04 3.202847e-05 -1.791988e-04 0.005522384 0
.004046689 0.019442560
C 0.0011722424 2.711960e-04 -1.643575e-05 2.464993e-05 -0.002186109 0
.002454132 -0.008387581
D -0.0015899760 9.948438e-05 -6.790312e-05 -5.471208e-04 -0.006243753 0
.003994443 0.010726061
E 0.0009475916 -5.727850e-05 7.624242e-05 6.062901e-04 0.011859843 -0
.008821349 -0.002486073
magnet_belt_x magnet_belt_y magnet_belt_z roll_arm pitch_arm ya
w_arm total_accel_arm
B -0.021459955 0.0280579103 0.071911618 0.003798006 -0.06576813 -0.0250
97131 -0.0208380039
C 0.007047188 -0.0007373633 -0.008946121 0.019143671 0.03876987 0.0239
03727 -0.0117818993
D -0.004494439 0.0014234456 -0.001797440 -0.002427299 -0.01689625 -0.0017
80834 0.0003814782
E 0.025647016 -0.0056357674 0.016470815 0.019009746 0.01069149 0.0230
81535 -0.0010059266
gyros_arm_x gyros_arm_y gyros_arm_z accel_arm_x accel_arm_y accel_
arm_z magnet_arm_y
```

```

B -0.0136631649 7.976717e-03 0.0001970857 -0.05704138 0.021008371 0.0434
76960 0.031921773
C 0.0025245032 -2.212564e-03 0.0002045377 -0.01685926 -0.004174773 0.0078
57859 -0.051624345
D 0.0004537182 1.180763e-05 -0.0002996302 0.03288859 -0.001635021 -0.0165
28067 -0.018772245
E 0.0008257342 -6.560845e-04 0.0002976939 -0.03988844 0.022885775 0.0372
78992 -0.002718483
magnet_arm_z roll_dumbbell pitch_dumbbell yaw_dumbbell gyros_dumbbell_x gy
ros_dumbbell_z
B -0.06237601 -0.007141826 -0.020049687 0.008375728 -0.0008487920
0.0016483160
C 0.01507292 -0.004422265 0.004605364 0.005811493 0.0003230252
-0.0012762521
D 0.02225466 0.023570751 0.010568385 0.027368428 -0.0001990213
-0.0003307281
E -0.03772053 -0.030077984 -0.017389277 -0.021913405 0.0002969471
0.0001542316
accel_dumbbell_x accel_dumbbell_y accel_dumbbell_z magnet_dumbbell_x magne
t_dumbbell_y
B 0.004025183 0.01887781 0.01304531 0.006073832
-0.024380679
C -0.011994461 -0.03696102 0.03148393 -0.011762556
-0.020482381
D -0.006274208 -0.02829597 -0.01092769 -0.022455773
0.048481796
E 0.035197146 0.04121934 0.03266806 -0.019823353
-0.005658537
magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm total_accel_fore
arm gyros_forearm_x
B -0.004437042 0.023174767 0.0004713158 -0.016731752 0.020793
016 2.453234e-03
C -0.010555326 0.041341274 0.0003490628 -0.004882358 0.012333
997 6.541527e-05
D -0.018808938 -0.003670976 0.0138553269 0.018909946 0.007584
437 1.026190e-04
E 0.036497263 0.007166679 0.0013250337 0.016361177 -0.012458
468 -4.443229e-04
gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y accel_fore
arm_z magnet_forearm_x
B -1.096523e-02 -1.697679e-03 0.004155049 0.03401914 0.0041
83016 0.015509573
C -5.693345e-05 -9.110254e-05 0.029864708 0.02577970 0.0025
81164 0.017710500
D 7.816097e-04 1.800418e-04 -0.040467072 -0.01966517 0.0081
47917 -0.010237055
E -1.034282e-04 7.203492e-05 0.019704081 0.02519823 0.0212
96631 -0.003299594
magnet_forearm_y magnet_forearm_z
B -0.03992453 0.0463339909
C -0.04187582 0.0488198263
D -0.01242610 0.0110447932
E -0.01985137 -0.0003666157

```

Std. Errors:

	(Intercept)	raw_timestamp_part_1	num_window	roll_belt	pitch_belt	yaw_belt	
B	1.289777e-19	1.706501e-10	9.503846e-18	1.627086e-17	3.507766e-18	1.004031e-18	
C	2.457477e-19	3.251476e-10	2.141019e-17	3.013940e-17	6.333887e-18	7.733794e-19	
D	1.215895e-18	1.608732e-09	3.226374e-17	8.186366e-19	7.674979e-18	1.133883e-16	
E	1.176744e-18	1.556938e-09	5.258754e-17	7.251690e-17	1.773989e-17	5.968324e-17	
	total_accel_belt	gyros_belt_x	gyros_belt_y	gyros_belt_z	accel_belt_x	accel_belt_y	accel_belt_z
B	2.590896e-18	5.886290e-20	4.532352e-21	5.873163e-20	5.471791e-18	8.906424e-18	2.323839e-17
C	4.798937e-18	1.162626e-19	7.216148e-21	1.115351e-19	1.016891e-17	1.743652e-17	4.253783e-17
D	3.587810e-18	1.105971e-20	1.850568e-20	1.345357e-19	2.103331e-17	1.482614e-18	2.389211e-17
E	1.291258e-17	2.916276e-19	1.565030e-20	7.216347e-19	3.459927e-17	4.549685e-17	9.785106e-17
	magnet_belt_x	magnet_belt_y	magnet_belt_z	roll_arm	pitch_arm	yaw_arm	total_accel_arm
B	3.022570e-18	7.393457e-17	4.997686e-17	5.546128e-17	6.704723e-18	8.430989e-17	2.831770e-18
C	3.491811e-19	1.436349e-16	9.066805e-17	2.717741e-17	3.391661e-18	1.218772e-17	6.081187e-18
D	5.165961e-19	7.294147e-16	3.779746e-16	7.955097e-17	3.819338e-17	4.637900e-17	1.974668e-17
E	1.292492e-17	6.783128e-16	4.534427e-16	3.481909e-17	2.546126e-17	8.362140e-17	3.405447e-17
	gyros_arm_x	gyros_arm_y	gyros_arm_z	accel_arm_x	accel_arm_y	accel_arm_z	magnet_arm_y
B	9.098726e-19	4.398910e-19	3.790563e-20	3.406499e-17	2.437789e-17	4.092270e-17	6.069455e-17
C	8.190068e-20	1.283379e-19	2.117975e-20	3.889764e-17	3.617581e-18	7.966507e-18	5.111921e-17
D	1.701341e-18	5.079584e-19	3.522974e-19	1.510558e-17	1.056285e-16	4.273232e-17	6.369871e-19
E	1.100189e-18	6.683111e-19	6.916372e-19	6.585145e-17	1.910074e-17	1.414372e-16	2.086086e-16
	magnet_arm_z	roll_dumbbell	pitch_dumbbell	yaw_dumbbell	gyros_dumbbell_x	gyros_dumbbell_z	
B	8.232175e-17	1.070253e-17	9.512748e-18	2.115266e-17	6.521144e-20	4.000531e-20	
C	6.588564e-17	1.423756e-17	2.307518e-18	1.984333e-17	1.384238e-19	1.034230e-19	
D	4.151328e-16	4.319632e-17	1.132785e-17	1.148090e-16	6.019352e-20	2.340041e-20	
E	3.387306e-16	7.416724e-17	7.384508e-17	6.434542e-17	8.611212e-20	6.020422e-21	
	accel_dumbbell_x	accel_dumbbell_y	accel_dumbbell_z	magnet_dumbbell_x	magnet_dumbbell_y		
B	9.857717e-18	1.119280e-17	1.209581e-17	8.781590e-17	6.346295e-17		
C	4.338462e-18	1.524084e-17	1.590679e-17	8.537480e-17	1.520607e-16		

```

D      8.378879e-18      2.333674e-17      2.476604e-17      6.271992e-16
3.956658e-16
E      1.350387e-16      1.005679e-16      3.640172e-17      4.319807e-16
1.237012e-16
  magnet_dumbbell_z roll_forearm pitch_forearm yaw_forearm total_accel_fore
arm gyros_forearm_x
B      1.610612e-17 2.202720e-17 1.103112e-17 1.380901e-17      4.246450e
-18      2.668999e-19
C      3.576154e-17 1.602152e-17 4.297630e-18 1.188324e-17      9.409439e
-18      1.855843e-20
D      5.877386e-18 8.511953e-17 4.837392e-17 7.485575e-17      4.345010e
-17      2.933953e-20
E      6.799727e-17 1.107671e-16 5.238683e-17 1.906894e-17      4.302260e
-17      3.149971e-19
  gyros_forearm_y gyros_forearm_z accel_forearm_x accel_forearm_y accel_fore
arm_z magnet_forearm_x
B      1.328172e-18 2.902488e-19      5.022358e-17      7.569670e-17      2.19505
7e-17      8.864237e-17
C      1.844195e-19 8.866300e-20      1.547405e-17      4.174947e-17      4.42071
2e-17      6.386250e-17
D      1.016503e-18 3.094208e-19      1.434339e-16      2.539318e-16      2.24280
3e-16      6.590090e-16
E      1.951326e-18 5.880521e-19      2.938107e-16      2.678555e-16      2.66217
2e-16      8.307565e-17
  magnet_forearm_y magnet_forearm_z
B      2.788504e-16 1.339348e-16
C      9.695325e-17 1.430405e-16
D      3.833278e-16 6.270757e-16
E      9.354359e-16 5.448831e-16

Residual Deviance: 98.86039
AIC: 514.8604
> # Predictions
> predicted <- predict(final, newdata= test)
> # b. Goodness of Fit
> library(car)
> chisq.test(table(test$classe), prop.table(table(predicted)))

      Pearson's Chi-squared test

data:  table(test$classe) and prop.table(table(predicted))
X-squared = 20, df = 16, p-value = 0.2202

Warning message:
In chisq.test(table(test$classe), prop.table(table(predicted))) :
  Chi-squared approximation may be incorrect
> # -----
--
> # c. Report the accuracy measures
> # Accuracy
> conf <- table(test$classe, predicted)
> OAA <- (conf[1,1]+conf[2,2]+conf[3,3]+conf[4,4]+conf[5,5]) / sum(conf)
> OAA
[1] 0.9900662

```

```
# -----
```

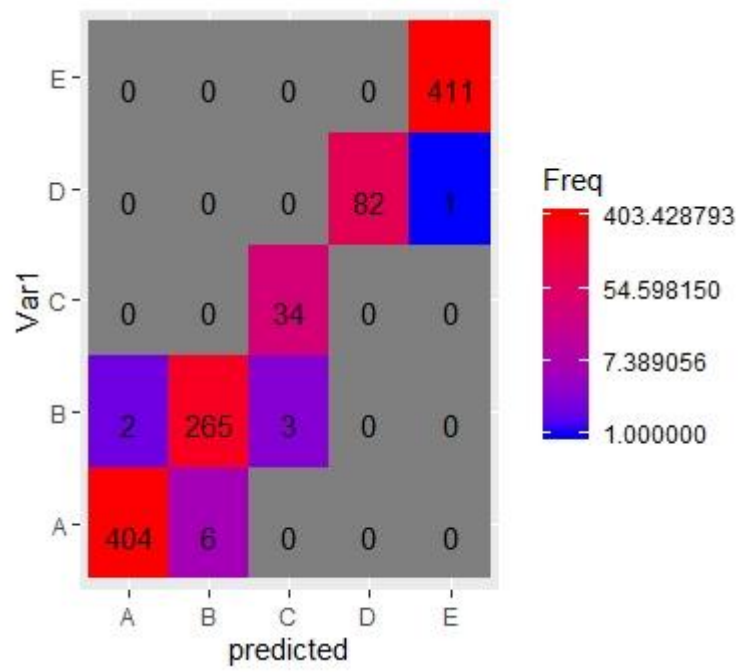
```
# g. Visualize the results
```

```
plot <- plot(conf, col = topo.colors(6))
```

```
library(ggplot2)
```

```
ggplot(data = as.data.frame(conf), mapping = aes(x = predicted,y = Var1)) +  
  geom_tile(aes(fill = Freq)) +  
  geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +  
  scale_fill_gradient(low = "blue", high = "red", trans = "log")
```

```
> library(pROC)  
Type 'citation("pROC")' for a citation.  
  
Attaching package: 'pROC'  
  
The following object is masked from 'package:glmnet':  
  
    auc  
  
The following objects are masked from 'package:stats':  
  
    cov, smooth, var  
  
Warning message:  
package 'pROC' was built under R version 3.6.1  
> m <- multiclass.roc(as.numeric(classe) ~ as.numeric(predicted) , data = te  
st)  
Error in multiclass.roc.formula(as.numeric(classe) ~ as.numeric(predicted),  
 :  
  Error in the formula: a response is required in a formula of type response  
~predictor.  
In addition: Warning message:  
In eval(predvars, data, env) : NAs introduced by coercion  
> plot <- plot(conf, col = topo.colors(6))  
Error in plot.new() : figure margins too large  
> library(ggplot2)  
> ggplot(data = as.data.frame(conf), mapping = aes(x = predicted,y = Var1))  
+  
+   geom_tile(aes(fill = Freq)) +  
+   geom_text(aes(label = sprintf("%1.0f", Freq)), vjust = 1) +  
+   scale_fill_gradient(low = "blue", high = "red", trans = "log")  
Warning message:  
Transformation introduced infinite values in discrete y-axis
```



Task 3:

1. Use the below-given data set

DataSet

2. Perform the below-given activities:

a. Create a classification model using different decision trees.

b. Verify model goodness of fit.

c. Apply all the model validation techniques.

d. Make conclusions

```
View(weight_lifting_exercises)
str(weight_lifting_exercises)
weight_lifting_exercises<-data.frame(weight_lifting_exercises[,
c(11:35,49:58,68:82,86:100,102:111,124:138,140:149)])
```

```
str(weight_lifting_exercises)
summary(weight_lifting_exercises)
```

```
weightTrain<-weight_lifting_exercises[1:2012,]
weightTest<-weight_lifting_exercises[2013:4024,]
summary(weightTrain)
names(weightTrain)
```

```
> #task18
> View(weight_lifting_exercises)
> str(weight_lifting_exercises)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    4024 obs. of  1
59 variables:
 $ user_name           : chr  "eurico" "eurico" "eurico" "eurico" ...
 $ raw_timestamp_part_1 : num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+
09 ...
 $ raw_timestamp_part_2 : num  34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp      : chr  "28/11/2011 14:15" "28/11/2011 14:15" "28/
11/2011 14:15" "28/11/2011 14:15" ...
 $ new_window          : chr  "no" "no" "no" "no" ...
 $ num_window          : num  1 1 1 1 1 1 1 1 1 1 ...
 $ roll_belt           : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.3
1 2 ...
```

```

$ pitch_belt      : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47
.4 47.7 ...
$ yaw_belt        : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9
-82.2 -82.6 -82.8 ...
$ total_accel_belt : num   3 2 1 1 1 1 3 4 2 3 ...
$ kurtosis_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_belt : chr  NA NA NA NA ...
$ kurtosis_yaw_belt  : chr  NA NA NA NA ...
$ skewness_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ skewness_roll_belt_1 : chr  NA NA NA NA ...
$ skewness_yaw_belt  : chr  NA NA NA NA ...
$ max_roll_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ gyros_belt_x       : num   2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.6
5 1.48 ...
$ gyros_belt_y       : num   0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0
.03 -0.06 ...
$ gyros_belt_z       : num   0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0
33 0.21 ...
$ accel_belt_x       : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y       : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z       : num   22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x      : num  387 405 409 422 418 432 438 440 443 449 ..
.
$ magnet_belt_y      : num  525 512 511 513 508 510 508 503 507 499 ..
.
$ magnet_belt_z      : num  -267 -254 -244 -221 -208 -189 -176 -163 -1
40 -132 ...
$ roll_arm          : num   132 129 125 120 115 110 104 98.6 93.2 88.5
...
$ pitch_arm         : num  -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9
-49.7 -49 -48.1 ...
$ yaw_arm           : num  -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -1
1.4 -4.49 1.82 ...
$ total_accel_arm    : num   38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm    : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA ...

```

```

$ avg_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x        : num  2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4
3.48 ...
$ gyros_arm_y        : num  -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83
-0.83 -0.83 -0.8 ...
$ gyros_arm_z        : num  -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -
0.21 -0.11 -0.15 ...
$ accel_arm_x        : num  143 146 156 158 163 160 165 153 143 135 ..
.
$ accel_arm_y        : num  30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z        : num  -346 -339 -307 -305 -288 -274 -225 -218 -2
05 -134 ...
$ magnet_arm_x       : num  556 599 613 646 670 696 721 725 740 741 ..
.
$ magnet_arm_y       : num  -205 -206 -198 -186 -175 -174 -161 -152 -1
33 -115 ...
$ magnet_arm_z       : num  -374 -335 -319 -268 -241 -193 -121 -105 -4
3 14 ...
$ kurtosis_roll_arm  : chr  NA NA NA NA ...
$ kurtosis_pitch_arm : chr  NA NA NA NA ...
$ kurtosis_yaw_arm   : chr  NA NA NA NA ...
$ skewness_roll_arm  : chr  NA NA NA NA ...
$ skewness_pitch_arm : chr  NA NA NA NA ...
$ skewness_yaw_arm   : chr  NA NA NA NA ...
$ max_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell      : num  51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell     : num  11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell       : num  104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_yaw_dumbbell : chr  NA NA NA NA ...
$ skewness_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell : chr  NA NA NA NA ...
$ max_roll_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]

```

```

- attr(*, "spec")=
.. cols(
..   user_name = col_character(),
..   raw_timestamp_part_1 = col_double(),
..   raw_timestamp_part_2 = col_double(),
..   cvtd_timestamp = col_character(),
..   new_window = col_character(),
..   num_window = col_double(),
..   roll_belt = col_double(),
..   pitch_belt = col_double(),
..   yaw_belt = col_double(),
..   total_accel_belt = col_double(),
..   kurtosis_roll_belt = col_double(),
..   kurtosis_pitch_belt = col_character(),
..   kurtosis_yaw_belt = col_character(),
..   skewness_roll_belt = col_double(),
..   skewness_roll_belt_1 = col_character(),
..   skewness_yaw_belt = col_character(),
..   max_roll_belt = col_double(),
..   max_pitch_belt = col_double(),
..   max_yaw_belt = col_double(),
..   min_roll_belt = col_double(),
..   min_pitch_belt = col_double(),
..   min_yaw_belt = col_double(),
..   amplitude_roll_belt = col_double(),
..   amplitude_pitch_belt = col_double(),
..   amplitude_yaw_belt = col_double(),
..   var_total_accel_belt = col_double(),
..   avg_roll_belt = col_double(),
..   stddev_roll_belt = col_double(),
..   var_roll_belt = col_double(),
..   avg_pitch_belt = col_double(),
..   stddev_pitch_belt = col_double(),
..   var_pitch_belt = col_double(),
..   avg_yaw_belt = col_double(),
..   stddev_yaw_belt = col_double(),
..   var_yaw_belt = col_double(),
..   gyros_belt_x = col_double(),
..   gyros_belt_y = col_double(),
..   gyros_belt_z = col_double(),
..   accel_belt_x = col_double(),
..   accel_belt_y = col_double(),
..   accel_belt_z = col_double(),
..   magnet_belt_x = col_double(),
..   magnet_belt_y = col_double(),
..   magnet_belt_z = col_double(),
..   roll_arm = col_double(),
..   pitch_arm = col_double(),
..   yaw_arm = col_double(),
..   total_accel_arm = col_double(),
..   var_accel_arm = col_double(),
..   avg_roll_arm = col_double(),
..   stddev_roll_arm = col_double(),
..   var_roll_arm = col_double(),
..   avg_pitch_arm = col_double(),
..   stddev_pitch_arm = col_double(),

```



```
.. var_pitch_arm = col_double(),
.. avg_yaw_arm = col_double(),
.. stddev_yaw_arm = col_double(),
.. var_yaw_arm = col_double(),
.. gyros_arm_x = col_double(),
.. gyros_arm_y = col_double(),
.. gyros_arm_z = col_double(),
.. accel_arm_x = col_double(),
.. accel_arm_y = col_double(),
.. accel_arm_z = col_double(),
.. magnet_arm_x = col_double(),
.. magnet_arm_y = col_double(),
.. magnet_arm_z = col_double(),
.. kurtosis_roll_arm = col_character(),
.. kurtosis_pitch_arm = col_character(),
.. kurtosis_yaw_arm = col_character(),
.. skewness_roll_arm = col_character(),
.. skewness_pitch_arm = col_character(),
.. skewness_yaw_arm = col_character(),
.. max_roll_arm = col_double(),
.. max_pitch_arm = col_double(),
.. max_yaw_arm = col_double(),
.. min_roll_arm = col_double(),
.. min_pitch_arm = col_double(),
.. min_yaw_arm = col_double(),
.. amplitude_roll_arm = col_double(),
.. amplitude_pitch_arm = col_double(),
.. amplitude_yaw_arm = col_double(),
.. roll_dumbbell = col_double(),
.. pitch_dumbbell = col_double(),
.. yaw_dumbbell = col_double(),
.. kurtosis_roll_dumbbell = col_double(),
.. kurtosis_pitch_dumbbell = col_double(),
.. kurtosis_yaw_dumbbell = col_character(),
.. skewness_roll_dumbbell = col_double(),
.. skewness_pitch_dumbbell = col_double(),
.. skewness_yaw_dumbbell = col_character(),
.. max_roll_dumbbell = col_double(),
.. max_pitch_dumbbell = col_double(),
.. max_yaw_dumbbell = col_double(),
.. min_roll_dumbbell = col_double(),
.. min_pitch_dumbbell = col_double(),
.. min_yaw_dumbbell = col_double(),
.. amplitude_roll_dumbbell = col_double(),
.. amplitude_pitch_dumbbell = col_double(),
.. amplitude_yaw_dumbbell = col_double(),
.. total_accel_dumbbell = col_double(),
.. var_accel_dumbbell = col_double(),
.. avg_roll_dumbbell = col_double(),
.. stddev_roll_dumbbell = col_double(),
.. var_roll_dumbbell = col_double(),
.. avg_pitch_dumbbell = col_double(),
.. stddev_pitch_dumbbell = col_double(),
.. var_pitch_dumbbell = col_double(),
.. avg_yaw_dumbbell = col_double(),
.. stddev_yaw_dumbbell = col_double(),
```

```

..   var_yaw_dumbbell = col_double(),
..   gyros_dumbbell_x = col_double(),
..   gyros_dumbbell_y = col_double(),
..   gyros_dumbbell_z = col_double(),
..   accel_dumbbell_x = col_double(),
..   accel_dumbbell_y = col_double(),
..   accel_dumbbell_z = col_double(),
..   magnet_dumbbell_x = col_double(),
..   magnet_dumbbell_y = col_double(),
..   magnet_dumbbell_z = col_double(),
..   roll_forearm = col_double(),
..   pitch_forearm = col_double(),
..   yaw_forearm = col_double(),
..   kurtosis_roll_forearm = col_character(),
..   kurtosis_pitch_forearm = col_character(),
..   kurtosis_yaw_forearm = col_character(),
..   skewness_roll_forearm = col_character(),
..   skewness_pitch_forearm = col_character(),
..   skewness_yaw_forearm = col_character(),
..   max_roll_forearm = col_double(),
..   max_pitch_forearm = col_double(),
..   max_yaw_forearm = col_character(),
..   min_roll_forearm = col_double(),
..   min_pitch_forearm = col_double(),
..   min_yaw_forearm = col_character(),
..   amplitude_roll_forearm = col_double(),
..   amplitude_pitch_forearm = col_double(),
..   amplitude_yaw_forearm = col_character(),
..   total_accel_forearm = col_double(),
..   var_accel_forearm = col_double(),
..   avg_roll_forearm = col_double(),
..   stddev_roll_forearm = col_double(),
..   var_roll_forearm = col_double(),
..   avg_pitch_forearm = col_double(),
..   stddev_pitch_forearm = col_double(),
..   var_pitch_forearm = col_double(),
..   avg_yaw_forearm = col_double(),
..   stddev_yaw_forearm = col_double(),
..   var_yaw_forearm = col_double(),
..   gyros_forearm_x = col_double(),
..   gyros_forearm_y = col_double(),
..   gyros_forearm_z = col_double(),
..   accel_forearm_x = col_double(),
..   accel_forearm_y = col_double(),
..   accel_forearm_z = col_double(),
..   magnet_forearm_x = col_double(),
..   magnet_forearm_y = col_double(),
..   magnet_forearm_z = col_double(),
..   classe = col_character()
.. )
> weight_lifting_exercises<-data.frame(weight_lifting_exercises[,-c(11:35,49
:58,68:82,86:100,102:111,124:138,140:149)])
> str(weight_lifting_exercises)
'data.frame':   4024 obs. of  59 variables:
 $ user_name      : chr  "eurico" "eurico" "eurico" "eurico" ...

```

```

$ raw_timestamp_part_1: num 1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+09 .
..
$ raw_timestamp_part_2: num 34670 62641 70653 82654 90637 ...
$ cvtd_timestamp      : chr "28/11/2011 14:15" "28/11/2011 14:15" "28/11/2011 14:15" "28/11/2011 14:15" ...
$ new_window          : chr "no" "no" "no" "no" ...
$ num_window          : num 1 1 1 1 1 1 1 1 1 1 ...
$ roll_belt           : num 3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.31 2
...
$ pitch_belt          : num 41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47.4 47.7 ...
$ yaw_belt            : num -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9 -82.2 -82.6 -82.8 ...
$ total_accel_belt    : num 3 2 1 1 1 1 3 4 2 3 ...
$ gyros_belt_x        : num 2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.65 1.48 ...
$ gyros_belt_y        : num 0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0.03 -0.06 ...
$ gyros_belt_z        : num 0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.33 0.21 ...
$ accel_belt_x        : num -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y        : num -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z        : num 22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x       : num 387 405 409 422 418 432 438 440 443 449 ...
$ magnet_belt_y       : num 525 512 511 513 508 510 508 503 507 499 ...
$ magnet_belt_z       : num -267 -254 -244 -221 -208 -189 -176 -163 -140 -132 ...
$ roll_arm            : num 132 129 125 120 115 110 104 98.6 93.2 88.5 ...
$ pitch_arm           : num -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9 -49.7 -49 -48.1 ...
$ yaw_arm             : num -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -11.4 -4.49 1.82 ...
$ total_accel_arm     : num 38 38 35 35 34 33 29 28 27 22 ...
$ gyros_arm_x         : num 2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4 3.48 ...
$ gyros_arm_y         : num -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83 -0.83 -0.83 -0.83 ...
$ gyros_arm_z         : num -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -0.21 -0.11 -0.15 ...
$ accel_arm_x         : num 143 146 156 158 163 160 165 153 143 135 ...
$ accel_arm_y         : num 30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z         : num -346 -339 -307 -305 -288 -274 -225 -218 -205 -134 ...
$ magnet_arm_x        : num 556 599 613 646 670 696 721 725 740 741 ...
$ magnet_arm_y        : num -205 -206 -198 -186 -175 -174 -161 -152 -133 -115 ...
$ magnet_arm_z        : num -374 -335 -319 -268 -241 -193 -121 -105 -43 14 ...
$ roll_dumbbell       : num 51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell      : num 11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell        : num 104.3 100.2 101.1 99.8 100.4 ...
$ total_accel_dumbbell : num 4 4 4 5 4 4 4 4 4 4 ...
$ gyros_dumbbell_x    : num -0.31 -0.31 -0.31 -0.31 -0.31 -0.31 -0.31 -0.29 -0.29 -0.29 ...
$ gyros_dumbbell_y    : num 0.16 0.14 0.16 0.16 0.14 0.14 0.14 0.14 0.14 0.14 ...

```

```

$ gyros_dumbbell_z      : num  0.08 0.07 0.05 0.07 0.07 0.07 0.07 0.07 0.08 0
.07 ...
$ accel_dumbbell_x      : num   5  4  3  5  5  6  6  3  6  5 ...
$ accel_dumbbell_y      : num  21 22 23 24 23 22 22 23 22 21 ...
$ accel_dumbbell_z      : num  37 35 37 38 37 36 35 35 36 34 ...
$ magnet_dumbbell_x     : num -471 -472 -468 -469 -468 -473 -471 -476 -474 -
470 ...
$ magnet_dumbbell_y     : num  191 184 190 184 189 188 183 184 186 188 ...
$ magnet_dumbbell_z     : num  277 281 275 285 292 278 284 283 278 283 ...
$ roll_forearm          : num  -111 -112 -114 -115 -117 -118 -120 -122 -124 -
126 ...
$ pitch_forearm         : num   26.5 26.2 26 25.8 25.5 25.1 24.8 24.5 24 23.4
...
$ yaw_forearm           : num  138 138 137 137 137 137 136 135 135 134 ...
$ total_accel_forearm   : num   30 31 32 33 34 36 37 37 38 39 ...
$ gyros_forearm_x       : num  -0.05 -0.06 -0.05 0.02 0.08 0.14 0.21 0.27 0.3
7 0.43 ...
$ gyros_forearm_y       : num  -0.37 -0.37 -0.27 -0.24 -0.27 -0.29 -0.24 -0.1
8 -0.21 -0.26 ...
$ gyros_forearm_z       : num  -0.43 -0.59 -0.72 -0.79 -0.82 -0.82 -0.84 -0.8
5 -0.87 -0.89 ...
$ accel_forearm_x       : num  -170 -178 -182 -185 -188 -208 -215 -212 -221 -
220 ...
$ accel_forearm_y       : num   155 164 172 182 195 207 214 221 223 239 ...
$ accel_forearm_z       : num   184 182 185 188 188 190 190 190 194 196 ...
$ magnet_forearm_x      : num  -1160 -1150 -1130 -1120 -1100 -1090 -1060 -106
0 -1040 -1020 ...
$ magnet_forearm_y      : num   1400 1410 1400 1400 1400 1400 1410 1410 1400 1
410 ...
$ magnet_forearm_z      : num  -876 -871 -863 -855 -843 -838 -831 -829 -819 -
804 ...
$ classe                : chr   "E" "E" "E" "E" ...
> summary(weight_lifting_exercises)
  user_name      raw_timestamp_part_1 raw_timestamp_part_2 cvtd_timestamp
  new_window
Length:4024      Min.      :1.322e+09      Min.      :    297      Length:4024
  Length:4024
Class :character 1st Qu.:1.323e+09      1st Qu.:244321      Class :charact
er Class :character
Mode :character  Median :1.323e+09      Median :492342      Mode :charact
er Mode :character
                Mean      :1.323e+09      Mean      :490377
                3rd Qu.:1.323e+09      3rd Qu.:736278
                Max.      :1.323e+09      Max.      :996453

  num_window      roll_belt      pitch_belt      yaw_belt      total_
accel_belt
Min.      : 1.00      Min.      : -28.90      Min.      : -56.20      Min.      : -179.000      Min.
: 0.00
1st Qu.:24.00      1st Qu.:  1.38      1st Qu.:  6.22      1st Qu.: -93.100      1st Qu
.: 3.00
Median :46.00      Median :122.00      Median : 25.50      Median :  -4.940      Median
:19.00

```

Mean :46.33	Mean : 73.31	Mean : 14.16	Mean : -30.975	Mean
:12.77				
3rd Qu.:69.00	3rd Qu.:124.00	3rd Qu.: 26.40	3rd Qu.: -2.695	3rd Qu
.:20.00				
Max. :91.00	Max. :159.00	Max. : 60.30	Max. : 179.000	Max.
:26.00				
gyros_belt_x	gyros_belt_y	gyros_belt_z	accel_belt_x	
accel_belt_y				
Min. :-0.7900	Min. :-0.470000	Min. :-0.7700	Min. :-120.00	M
in. :-71.00				
1st Qu.: -0.4300	1st Qu.: -0.030000	1st Qu.: -0.4600	1st Qu.: -42.00	1
st Qu.: 4.00				
Median : -0.2400	Median : -0.020000	Median : -0.4100	Median : -34.00	M
edian : 65.00				
Mean :-0.1823	Mean :-0.008837	Mean :-0.2464	Mean : -24.36	M
ean : 39.84				
3rd Qu.: 0.0200	3rd Qu.: 0.000000	3rd Qu.: -0.0200	3rd Qu.: -16.00	3
rd Qu.: 70.00				
Max. : 2.0200	Max. : 0.420000	Max. : 0.8200	Max. : 80.00	M
ax. :164.00				
accel_belt_z	magnet_belt_x	magnet_belt_y	magnet_belt_z	roll
_arm				
Min. :-244.00	Min. :-30.00	Min. :428.0	Min. :-513.0	Min.
:-180.00				
1st Qu.: -176.00	1st Qu.: -3.00	1st Qu.:577.0	1st Qu.: -379.0	1st Qu.
: -34.40				
Median : -166.00	Median : 2.00	Median :585.0	Median : -366.0	Median
: 72.10				
Mean : -94.73	Mean : 24.65	Mean :582.7	Mean : -340.9	Mean
: 40.01				
3rd Qu.: 20.00	3rd Qu.: 8.00	3rd Qu.:601.0	3rd Qu.: -311.0	3rd Qu.
: 124.00				
Max. : 77.00	Max. :485.00	Max. :652.0	Max. : 293.0	Max.
: 180.00				
pitch_arm	yaw_arm	total_accel_arm	gyros_arm_x	gyr
os_arm_y				
Min. :-87.100	Min. :-180.000	Min. : 1.00	Min. :-5.2000	Min.
:-3.4400				
1st Qu.: -32.200	1st Qu.: -59.675	1st Qu.:15.00	1st Qu.: -2.0925	1st
Qu.: -0.9200				
Median : -8.645	Median : 17.500	Median :25.00	Median : -0.0200	Medi
an : -0.0300				
Mean :-10.539	Mean : 2.768	Mean :24.89	Mean : -0.1852	Mean
:-0.1818				
3rd Qu.: 14.600	3rd Qu.: 72.825	3rd Qu.:34.00	3rd Qu.: 1.7000	3rd
Qu.: 0.5800				
Max. : 81.400	Max. : 180.000	Max. :59.00	Max. : 4.3400	Max.
: 2.4600				
gyros_arm_z	accel_arm_x	accel_arm_y	accel_arm_z	m
agnet_arm_x				
Min. :-2.17000	Min. :-346.00	Min. :-252.00	Min. :-538.00	Mi
n. :-515.0				
1st Qu.: -0.20000	1st Qu.: -88.00	1st Qu.: -21.00	1st Qu.: -124.00	1s
t Qu.: -332.0				
Median : 0.00000	Median : 24.00	Median : 22.00	Median : 6.00	Me
dian : 278.5				

Mean : 0.04444	Mean : 34.38	Mean : 26.87	Mean : -41.39	Me
an : 194.3				
3rd Qu.: 0.28000	3rd Qu.: 136.00	3rd Qu.: 96.25	3rd Qu.: 76.00	3r
d Qu.: 651.0				
Max. : 3.02000	Max. : 434.00	Max. : 229.00	Max. : 209.00	Ma
x. : 782.0				
magnet_arm_y	magnet_arm_z	roll_dumbbell	pitch_dumbbell	yaw
_dumbbell				
Min. : -392.0	Min. : -573.0	Min. : -152.782	Min. : -134.73	Min.
: -129.33				
1st Qu.: -13.0	1st Qu.: -1.0	1st Qu.: -34.657	1st Qu.: -12.93	1st
Qu.: 21.35				
Median : 267.0	Median : 431.0	Median : -2.295	Median : 14.48	Medi
an : 72.49				
Mean : 161.7	Mean : 253.2	Mean : 3.500	Mean : 5.18	Mean
: 55.66				
3rd Qu.: 348.0	3rd Qu.: 515.0	3rd Qu.: 58.014	3rd Qu.: 27.95	3rd
Qu.: 122.01				
Max. : 482.0	Max. : 647.0	Max. : 139.729	Max. : 97.28	Max.
: 152.92				
total_accel_dumbbell	gyros_dumbbell_x	gyros_dumbbell_y	gyros_dumbbell_z	
accel_dumbbell_x				
Min. : 1.00	Min. : -1.4300	Min. : -2.04000	Min. : -1.4600	
Min. : -237.000				
1st Qu.: 6.00	1st Qu.: -0.0200	1st Qu.: -0.27000	1st Qu.: -0.3300	
1st Qu.: -6.000				
Median : 9.00	Median : 0.3200	Median : -0.06000	Median : -0.1300	
Median : 11.000				
Mean : 12.02	Mean : 0.2487	Mean : -0.04674	Mean : -0.1337	
Mean : -7.091				
3rd Qu.: 14.00	3rd Qu.: 0.5300	3rd Qu.: 0.14000	3rd Qu.: 0.0500	
3rd Qu.: 23.000				
Max. : 37.00	Max. : 1.4800	Max. : 4.37000	Max. : 1.8900	
Max. : 217.000				
accel_dumbbell_y	accel_dumbbell_z	magnet_dumbbell_x	magnet_dumbbell_y	mag
net_dumbbell_z				
Min. : -163.00	Min. : -273.00	Min. : -638.00	Min. : -730.0	Min
: -262.00				
1st Qu.: -28.00	1st Qu.: 12.00	1st Qu.: -515.00	1st Qu.: -544.0	1st
Qu.: -101.00				
Median : -2.00	Median : 51.00	Median : 107.50	Median : -486.0	Med
ian : -59.00				
Mean : 12.83	Mean : 16.63	Mean : 10.55	Mean : -115.7	Mea
n : -41.12				
3rd Qu.: 47.00	3rd Qu.: 79.00	3rd Qu.: 506.00	3rd Qu.: 304.0	3rd
Qu.: 1.00				
Max. : 281.00	Max. : 122.00	Max. : 579.00	Max. : 618.0	Max
: 300.00				
roll_forearm	pitch_forearm	yaw_forearm	total_accel_forearm	gyr
os_forearm_x				
Min. : -180.0	Min. : -64.00	Min. : -180.00	Min. : 10.00	Min
: -1.8800				
1st Qu.: -115.0	1st Qu.: 0.00	1st Qu.: -106.00	1st Qu.: 30.00	1st
Qu.: -0.1400				
Median : 89.5	Median : 19.70	Median : 83.50	Median : 35.00	Med
ian : 0.0600				

Mean : 36.1	Mean : 18.57	Mean : 17.79	Mean : 34.38	Mean : 1.8100
n : 0.1076				
3rd Qu.: 136.0	3rd Qu.: 43.90	3rd Qu.: 108.00	3rd Qu.: 37.00	3rd Qu.: 0.4200
Max. : 180.0	Max. : 86.90	Max. : 180.00	Max. : 59.00	Max. : 1.8100
gyros_forearm_y	gyros_forearm_z	accel_forearm_x	accel_forearm_y	
Min. : -5.730000	Min. : -2.580000	Min. : -328.000	Min. : -467.00	
Min. : -366				
1st Qu.: -1.780000	1st Qu.: -0.31000	1st Qu.: -117.000	1st Qu.: 75.75	
1st Qu.: -210				
Median : -0.020000	Median : -0.02000	Median : -6.000	Median : 229.50	
Median : -181				
Mean : -0.004108	Mean : 0.09302	Mean : -6.445	Mean : 171.47	
Mean : -163				
3rd Qu.: 1.830000	3rd Qu.: 0.48000	3rd Qu.: 113.000	3rd Qu.: 297.00	
3rd Qu.: -150				
Max. : 5.170000	Max. : 3.35000	Max. : 279.000	Max. : 575.00	
Max. : 239				
magnet_forearm_x	magnet_forearm_y	magnet_forearm_z	classe	
Min. : -1160.0	Min. : -725.0	Min. : -876.0	Length:4024	
1st Qu.: -589.0	1st Qu.: -76.0	1st Qu.: 370.8	Class :character	
Median : -330.5	Median : 653.0	Median : 560.0	Mode :character	
Mean : -348.7	Mean : 358.6	Mean : 475.2		
3rd Qu.: -152.0	3rd Qu.: 747.0	3rd Qu.: 670.0		
Max. : 413.0	Max. : 1440.0	Max. : 1040.0		

#Ques.2. Perform the below given activities:

a. Create classification model using different decision trees.

```
weightTrain<-data.frame(weightTrain[,-
c(11:35,49:58,68:82,86:100,102:111,124:138,140:149)])
library(caret)
library(Hmisc)
weightTrain$raw_timestamp_part_1<-impute(weightTrain$raw_timestamp_part_1,mean)
weightTrain$raw_timestamp_part_2<-impute(weightTrain$raw_timestamp_part_2,mean)
weightTrain$cvtd_timestamp<-impute(weightTrain$cvtd_timestamp,mean)
weightTrain$new_window<-impute(weightTrain$new_window,mean)
weightTrain$num_window<-impute(weightTrain$num_window,mean)
weightTrain$roll_belt<-impute(weightTrain$roll_belt,mean)
weightTrain$pitch_belt<-impute(weightTrain$pitch_belt,mean)
weightTrain$yaw_belt<-impute(weightTrain$yaw_belt,mean)
summary(weightTrain)
str(weightTrain)
.. )
> weight_lifting_exercises<-data.frame(weight_lifting_exercises[,-c(11:35,49
:58,68:82,86:100,102:111,124:138,140:149)])
> str(weight_lifting_exercises)
'data.frame': 4024 obs. of 59 variables:
```

```

$ user_name      : chr  "eurico" "eurico" "eurico" "eurico" ...
$ raw_timestamp_part_1: num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+09 .
..
$ raw_timestamp_part_2: num  34670 62641 70653 82654 90637 ...
$ cvtd_timestamp    : chr  "28/11/2011 14:15" "28/11/2011 14:15" "28/11/2
011 14:15" "28/11/2011 14:15" ...
$ new_window        : chr  "no" "no" "no" "no" ...
$ num_window         : num  1 1 1 1 1 1 1 1 1 1 ...
$ roll_belt          : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.31 2
...
$ pitch_belt         : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47.4 4
7.7 ...
$ yaw_belt           : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9 -82.
2 -82.6 -82.8 ...
$ total_accel_belt   : num  3 2 1 1 1 1 3 4 2 3 ...
$ gyros_belt_x        : num  2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.65 1.
48 ...
$ gyros_belt_y        : num  0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0.03
-0.06 ...
$ gyros_belt_z        : num  0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.33 0
.21 ...
$ accel_belt_x        : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y        : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z        : num  22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x       : num  387 405 409 422 418 432 438 440 443 449 ...
$ magnet_belt_y       : num  525 512 511 513 508 510 508 503 507 499 ...
$ magnet_belt_z       : num  -267 -254 -244 -221 -208 -189 -176 -163 -140 -
132 ...
$ roll_arm           : num  132 129 125 120 115 110 104 98.6 93.2 88.5 ...
$ pitch_arm          : num  -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9 -49.
7 -49 -48.1 ...
$ yaw_arm            : num  -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -11.4
-4.49 1.82 ...
$ total_accel_arm     : num  38 38 35 35 34 33 29 28 27 22 ...
$ gyros_arm_x         : num  2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4 3.48
...
$ gyros_arm_y         : num  -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83 -0.8
3 -0.83 -0.8 ...
$ gyros_arm_z         : num  -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -0.21
-0.11 -0.15 ...
$ accel_arm_x         : num  143 146 156 158 163 160 165 153 143 135 ...
$ accel_arm_y         : num  30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z         : num  -346 -339 -307 -305 -288 -274 -225 -218 -205 -
134 ...
$ magnet_arm_x        : num  556 599 613 646 670 696 721 725 740 741 ...
$ magnet_arm_y        : num  -205 -206 -198 -186 -175 -174 -161 -152 -133 -
115 ...
$ magnet_arm_z        : num  -374 -335 -319 -268 -241 -193 -121 -105 -43 14
...
$ roll_dumbbell       : num  51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell      : num  11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell        : num  104.3 100.2 101.1 99.8 100.4 ...
$ total_accel_dumbbell: num  4 4 4 5 4 4 4 4 4 4 ...
$ gyros_dumbbell_x     : num  -0.31 -0.31 -0.31 -0.31 -0.31 -0.31 -0.29 -0.2
9 -0.29 -0.29 ...

```



```

$ gyros_dumbbell_y : num 0.16 0.14 0.16 0.16 0.14 0.14 0.14 0.14 0.14 0
.14 ...
$ gyros_dumbbell_z : num 0.08 0.07 0.05 0.07 0.07 0.07 0.07 0.07 0.08 0
.07 ...
$ accel_dumbbell_x : num 5 4 3 5 5 6 6 3 6 5 ...
$ accel_dumbbell_y : num 21 22 23 24 23 22 22 23 22 21 ...
$ accel_dumbbell_z : num 37 35 37 38 37 36 35 35 36 34 ...
$ magnet_dumbbell_x : num -471 -472 -468 -469 -468 -473 -471 -476 -474 -
470 ...
$ magnet_dumbbell_y : num 191 184 190 184 189 188 183 184 186 188 ...
$ magnet_dumbbell_z : num 277 281 275 285 292 278 284 283 278 283 ...
$ roll_forearm : num -111 -112 -114 -115 -117 -118 -120 -122 -124 -
126 ...
$ pitch_forearm : num 26.5 26.2 26 25.8 25.5 25.1 24.8 24.5 24 23.4
...
$ yaw_forearm : num 138 138 137 137 137 137 136 135 135 134 ...
$ total_accel_forearm : num 30 31 32 33 34 36 37 37 38 39 ...
$ gyros_forearm_x : num -0.05 -0.06 -0.05 0.02 0.08 0.14 0.21 0.27 0.3
7 0.43 ...
$ gyros_forearm_y : num -0.37 -0.37 -0.27 -0.24 -0.27 -0.29 -0.24 -0.1
8 -0.21 -0.26 ...
$ gyros_forearm_z : num -0.43 -0.59 -0.72 -0.79 -0.82 -0.82 -0.84 -0.8
5 -0.87 -0.89 ...
$ accel_forearm_x : num -170 -178 -182 -185 -188 -208 -215 -212 -221 -
220 ...
$ accel_forearm_y : num 155 164 172 182 195 207 214 221 223 239 ...
$ accel_forearm_z : num 184 182 185 188 188 190 190 190 194 196 ...
$ magnet_forearm_x : num -1160 -1150 -1130 -1120 -1100 -1090 -1060 -106
0 -1040 -1020 ...
$ magnet_forearm_y : num 1400 1410 1400 1400 1400 1400 1410 1410 1400 1
410 ...
$ magnet_forearm_z : num -876 -871 -863 -855 -843 -838 -831 -829 -819 -
804 ...
$ classe : chr "E" "E" "E" "E" ...
> summary(weight_lifting_exercises)
  user_name      raw_timestamp_part_1 raw_timestamp_part_2 cvtd_timestamp
  new_window
Length:4024      Min.      :1.322e+09      Min.      : 297      Length:4024
  Length:4024
Class :character 1st Qu.:1.323e+09      1st Qu.:244321      Class :charact
er  Class :character
Mode :character  Median :1.323e+09      Median :492342      Mode :charact
er  Mode :character
              Mean      :1.323e+09      Mean      :490377
              3rd Qu.:1.323e+09      3rd Qu.:736278
              Max.      :1.323e+09      Max.      :996453

  num_window      roll_belt      pitch_belt      yaw_belt      total_
accel_belt
Min.      : 1.00      Min.      : -28.90      Min.      : -56.20      Min.      : -179.000      Min.
: 0.00
1st Qu.:24.00      1st Qu.: 1.38      1st Qu.: 6.22      1st Qu.: -93.100      1st Qu
.: 3.00

```

Median :46.00	Median :122.00	Median : 25.50	Median : -4.940	Median
:19.00				
Mean :46.33	Mean : 73.31	Mean : 14.16	Mean : -30.975	Mean
:12.77				
3rd Qu.:69.00	3rd Qu.:124.00	3rd Qu.: 26.40	3rd Qu.: -2.695	3rd Qu
.:20.00				
Max. :91.00	Max. :159.00	Max. : 60.30	Max. : 179.000	Max.
:26.00				
gyros_belt_x	gyros_belt_y	gyros_belt_z	accel_belt_x	
accel_belt_y				
Min. :-0.7900	Min. :-0.470000	Min. :-0.7700	Min. :-120.00	M
in. :-71.00				
1st Qu.: -0.4300	1st Qu.: -0.030000	1st Qu.: -0.4600	1st Qu.: -42.00	1
st Qu.: 4.00				
Median :-0.2400	Median :-0.020000	Median :-0.4100	Median : -34.00	M
edian : 65.00				
Mean :-0.1823	Mean :-0.008837	Mean :-0.2464	Mean : -24.36	M
ean : 39.84				
3rd Qu.: 0.0200	3rd Qu.: 0.000000	3rd Qu.: -0.0200	3rd Qu.: -16.00	3
rd Qu.: 70.00				
Max. : 2.0200	Max. : 0.420000	Max. : 0.8200	Max. : 80.00	M
ax. :164.00				
accel_belt_z	magnet_belt_x	magnet_belt_y	magnet_belt_z	roll
_arm				
Min. :-244.00	Min. :-30.00	Min. :428.0	Min. :-513.0	Min.
:-180.00				
1st Qu.: -176.00	1st Qu.: -3.00	1st Qu.:577.0	1st Qu.: -379.0	1st Qu.
: -34.40				
Median :-166.00	Median : 2.00	Median :585.0	Median :-366.0	Median
: 72.10				
Mean : -94.73	Mean : 24.65	Mean :582.7	Mean :-340.9	Mean
: 40.01				
3rd Qu.: 20.00	3rd Qu.: 8.00	3rd Qu.:601.0	3rd Qu.: -311.0	3rd Qu.
: 124.00				
Max. : 77.00	Max. :485.00	Max. :652.0	Max. : 293.0	Max.
: 180.00				
pitch_arm	yaw_arm	total_accel_arm	gyros_arm_x	gyr
os_arm_y				
Min. :-87.100	Min. :-180.000	Min. : 1.00	Min. :-5.2000	Min.
:-3.4400				
1st Qu.: -32.200	1st Qu.: -59.675	1st Qu.:15.00	1st Qu.: -2.0925	1st
Qu.: -0.9200				
Median : -8.645	Median : 17.500	Median :25.00	Median :-0.0200	Medi
an :-0.0300				
Mean :-10.539	Mean : 2.768	Mean :24.89	Mean :-0.1852	Mean
:-0.1818				
3rd Qu.: 14.600	3rd Qu.: 72.825	3rd Qu.:34.00	3rd Qu.: 1.7000	3rd
Qu.: 0.5800				
Max. : 81.400	Max. : 180.000	Max. :59.00	Max. : 4.3400	Max.
: 2.4600				
gyros_arm_z	accel_arm_x	accel_arm_y	accel_arm_z	m
agnet_arm_x				
Min. :-2.17000	Min. :-346.00	Min. :-252.00	Min. :-538.00	Mi
n. :-515.0				
1st Qu.: -0.20000	1st Qu.: -88.00	1st Qu.: -21.00	1st Qu.: -124.00	1s
t Qu.: -332.0				

Median : 0.00000	Median : 24.00	Median : 22.00	Median : 6.00	Median : 278.5
Mean : 0.04444	Mean : 34.38	Mean : 26.87	Mean : -41.39	Mean : 194.3
3rd Qu.: 0.28000	3rd Qu.: 136.00	3rd Qu.: 96.25	3rd Qu.: 76.00	3rd Qu.: 651.0
Max. : 3.02000	Max. : 434.00	Max. : 229.00	Max. : 209.00	Max. : 782.0
magnet_arm_y_dumbbell	magnet_arm_z	roll_dumbbell	pitch_dumbbell	yaw_dumbbell
Min. : -392.0	Min. : -573.0	Min. : -152.782	Min. : -134.73	Min. : -129.33
1st Qu.: -13.0	1st Qu.: -1.0	1st Qu.: -34.657	1st Qu.: -12.93	1st Qu.: 21.35
Median : 267.0	Median : 431.0	Median : -2.295	Median : 14.48	Median : 72.49
Mean : 161.7	Mean : 253.2	Mean : 3.500	Mean : 5.18	Mean : 55.66
3rd Qu.: 348.0	3rd Qu.: 515.0	3rd Qu.: 58.014	3rd Qu.: 27.95	3rd Qu.: 122.01
Max. : 482.0	Max. : 647.0	Max. : 139.729	Max. : 97.28	Max. : 152.92
total_accel_dumbbell	gyros_dumbbell_x	gyros_dumbbell_y	gyros_dumbbell_z	accel_dumbbell_x
Min. : 1.00	Min. : -1.4300	Min. : -2.04000	Min. : -1.4600	Min. : -237.000
1st Qu.: 6.00	1st Qu.: -0.0200	1st Qu.: -0.27000	1st Qu.: -0.3300	1st Qu.: -6.000
Median : 9.00	Median : 0.3200	Median : -0.06000	Median : -0.1300	Median : 11.000
Mean : 12.02	Mean : 0.2487	Mean : -0.04674	Mean : -0.1337	Mean : -7.091
3rd Qu.: 14.00	3rd Qu.: 0.5300	3rd Qu.: 0.14000	3rd Qu.: 0.0500	3rd Qu.: 23.000
Max. : 37.00	Max. : 1.4800	Max. : 4.37000	Max. : 1.8900	Max. : 217.000
accel_dumbbell_y	accel_dumbbell_z	magnet_dumbbell_x	magnet_dumbbell_y	magnet_dumbbell_z
Min. : -163.00	Min. : -273.00	Min. : -638.00	Min. : -730.0	Min. : -262.00
1st Qu.: -28.00	1st Qu.: 12.00	1st Qu.: -515.00	1st Qu.: -544.0	1st Qu.: -101.00
Median : -2.00	Median : 51.00	Median : 107.50	Median : -486.0	Median : -59.00
Mean : 12.83	Mean : 16.63	Mean : 10.55	Mean : -115.7	Mean : -41.12
3rd Qu.: 47.00	3rd Qu.: 79.00	3rd Qu.: 506.00	3rd Qu.: 304.0	3rd Qu.: 1.00
Max. : 281.00	Max. : 122.00	Max. : 579.00	Max. : 618.0	Max. : 300.00
roll_forearm	pitch_forearm	yaw_forearm	total_accel_forearm	gyros_forearm_x
Min. : -180.0	Min. : -64.00	Min. : -180.00	Min. : 10.00	Min. : -1.8800
1st Qu.: -115.0	1st Qu.: 0.00	1st Qu.: -106.00	1st Qu.: 30.00	1st Qu.: -0.1400

```

Median : 89.5    Median : 19.70    Median : 83.50    Median :35.00    Med
ian : 0.0600
Mean   : 36.1    Mean   : 18.57    Mean   : 17.79    Mean   :34.38    Mea
n     : 0.1076
3rd Qu.: 136.0   3rd Qu.: 43.90   3rd Qu.: 108.00   3rd Qu.:37.00   3rd
Qu.: 0.4200
Max.   : 180.0   Max.   : 86.90   Max.   : 180.00   Max.   :59.00   Max
.     : 1.8100
gyros_forearm_y    gyros_forearm_z    accel_forearm_x    accel_forearm_y
accel_forearm_z
Min.   :-5.730000   Min.   :-2.58000   Min.   :-328.000   Min.   :-467.00
Min.   :-366
1st Qu.: -1.780000   1st Qu.: -0.31000   1st Qu.: -117.000   1st Qu.: 75.75
1st Qu.: -210
Median : -0.020000   Median : -0.02000   Median : -6.000   Median : 229.50
Median : -181
Mean   :-0.004108   Mean   : 0.09302   Mean   : -6.445   Mean   : 171.47
Mean   :-163
3rd Qu.: 1.830000   3rd Qu.: 0.48000   3rd Qu.: 113.000   3rd Qu.: 297.00
3rd Qu.: -150
Max.   : 5.170000   Max.   : 3.35000   Max.   : 279.000   Max.   : 575.00
Max.   : 239
magnet_forearm_x    magnet_forearm_y    magnet_forearm_z    classe
Min.   :-1160.0   Min.   :-725.0   Min.   :-876.0   Length:4024
1st Qu.: -589.0   1st Qu.: -76.0   1st Qu.: 370.8   Class :character
Median : -330.5   Median : 653.0   Median : 560.0   Mode :character
Mean   : -348.7   Mean   : 358.6   Mean   : 475.2
3rd Qu.: -152.0   3rd Qu.: 747.0   3rd Qu.: 670.0
Max.   : 413.0   Max.   :1440.0   Max.   :1040.0
> # a. Create classification model using different decision trees.
> weightTrain<-data.frame(weightTrain[,-c(11:35,49:58,68:82,86:100,102:111,1
24:138,140:149)])
> library(caret)
> library(Hmisc)
> weightTrain$raw_timestamp_part_1<-impute(weightTrain$raw_timestamp_part_1,
mean)
> weightTrain$raw_timestamp_part_2<-impute(weightTrain$raw_timestamp_part_2,
mean)
> weightTrain$cvtd_timestamp<-impute(weightTrain$cvtd_timestamp,mean)
> weightTrain$new_window<-impute(weightTrain$new_window,mean)
> weightTrain$num_window<-impute(weightTrain$num_window,mean)
> weightTrain$roll_belt<-impute(weightTrain$roll_belt,mean)
> weightTrain$pitch_belt<-impute(weightTrain$pitch_belt,mean)
> weightTrain$yaw_belt<-impute(weightTrain$yaw_belt,mean)
> summary(weightTrain)
Error in if (all(oi == oi[1])) cat("\n", length(i), "values imputed to", :
missing value where TRUE/FALSE needed
> str(weightTrain)
'data.frame': 2012 obs. of 10 variables:
 $ user_name      : chr "eurico" "eurico" "eurico" "eurico" ...
 $ raw_timestamp_part_1: num 1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+09 .
..
 $ raw_timestamp_part_2: num 34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp   : 'impute' num NaN NaN NaN NaN NaN NaN NaN NaN NaN N
aN ...
..- attr(*, "imputed")= int 1 2 3 4 5 6 7 8 9 10 ...

```

```

$ new_window      : 'impute' num  NaN NaN NaN NaN NaN NaN NaN NaN NaN NaN N
aN ...
..- attr(*, "imputed")= int   1 2 3 4 5 6 7 8 9 10 ...
$ num_window      : num    1 1 1 1 1 1 1 1 1 1 ...
$ roll_belt       : num    3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.31 2
...
$ pitch_belt      : num    41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47.4 4
7.7 ...
$ yaw_belt        : num   -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9 -82.
2 -82.6 -82.8 ...
$ total_accel_belt : num    3 2 1 1 1 1 3 4 2 3 ...

```

```

weightTrain$cvtd_timestamp<-as.integer(weightTrain$cvtd_timestamp)
weightTrain$new_window<-as.integer(weightTrain$new_window)
library(tree)
tree<-tree(classe~. ,
            data = weightTrain)
plot(tree,pretty = 0.1)
text(tree,pretty = 1.2)
summary(tree)

```

```

library(caret)
pred <- predict(tree,weightTrain,type='class')
str(pred)
dim(pred)
dim(weightTest$classe)

```

```

weightTest$classe<-as.factor(weightTest$classe)
dim(weightTest$classe)
table(weightTest$classe,pred)

```

```

length(pred)
length(weightTest$classe)
confusionMatrix(pred,weightTest$classe)
#.....

```

```

install.packages("rpart")
library(rpart)
fit1 <- rpart(classe~.,data=weightTrain[,-1])
class(fit1)
summary(fit1)

```

```

> library(rpart)
Warning message:
package 'rpart' was built under R version 3.6.1
> fit1 <- rpart(classe~.,data=weightTrain[,-1])
Error in eval(predvars, data, env) : object 'classe' not found
> class(fit1)
[1] "cv.glmnet"

```

```

> summary(fit1)
      Length Class  Mode
lambda     100  -none- numeric
cvm         100  -none- numeric
cvstd       100  -none- numeric
cvup        100  -none- numeric
cvlo        100  -none- numeric
nzero       100  -none- numeric
name         1  -none- character
glmnet.fit   12  elnet  list
lambda.min    1  -none- numeric
lambda.1se    1  -none- numeric
> rpart.plot::rpart.plot(fit1)

```

```
rpart.plot::rpart.plot(fit1)
```

```

pred1<-predict(fit1,weightTrain,type = "class")
summary(pred1)

```

```

summary(pred1)
  A    D    E
366 270 1376
> dim(pred1)

```

```

dim(pred1)
weightTest$classe<-as.factor(weightTest$classe)
table(weightTest$classe,pred1)
confusionMatrix(weightTest$classe,pred1)

```

```

> weightTest$classe<-as.factor(weightTest$classe)
> table(weightTest$classe,pred1)
  pred1
    A    D    E
A 337 253 409
B   0   17 884
C  29   0  83

```

```

# b. Verify model goodness of fit.
#.....for pred.....
weightTest$classe<-as.factor(weightTest$classe)
dim(weightTest$classe)
table(weightTest$classe,pred)

```

```

length(pred)
length(weightTest$classe)
confusionMatrix(pred,weightTest$classe)

```

```
#...for fit1....
weightTest$classe<-as.factor(weightTest$classe)
table(weightTest$classe,pred1)
confusionMatrix(weightTest$classe,pred1)
```

```
> weightTest$classe<-as.factor(weightTest$classe)
> table(weightTest$classe,pred1)
  pred1
      A      D      E
A  337  253  409
B    0   17  884
C   29    0   83
```

c. Apply all the model validation techniques.

```
set.seed(3)
install.packages('tree')
library(tree)
cv.weight<-cv.tree(tree,FUN = prune.misclass) #cv->cross validation
cv.weight_lifting_exercises<-cv.tree(tree,FUN = prune.misclass)
names(cv.weight)
cv.weight
```

```
par(mfrow = c(1,2))
plot(cv.weight$size,cv.weight$dev,type = 'b',col = 'red')
```

```
prune.weight<-prune.misclass(tree,best = 9)
plot(prune.weight)
text(prune.weight,pretty = 0)
```

```
weightTrain$cvtd_timestamp<-as.integer(weightTrain$cvtd_timestamp)
weightTrain$new_window<-as.integer(weightTrain$new_window)
tree.pred1<-predict(prune.weight,weightTrain,type = 'class')
table(tree.pred1,weightTest)
```

```
#.....Random forest.....
library(randomForest)
set.seed(1)
a.weight_lifting_exercises<-randomForest(classe~.,weight_lifting_exercises,
                                          subset = weightTrain,mtry = 3,importance = TRUE)
dim(a.weight_lifting_exercises)

importance(a.weight_lifting_exercises)
```

```
varImpPlot(a.weight_lifting_exercises,col = 'blue',pch = 10, cex = 1.25)
```

```
a.weight_lifting_exercises
```

```
test.pred.rf<-predict(a.weight_lifting_exercises, newdata = weight_lifting_exercises[-  
weightTrain,],type = 'class')  
table(test.pred.rf,weightTest)
```

```
#.....adaboost.....
```

```
install.packages(adabag)  
library(adabag)  
set.seed(300)  
weight_lifting_exercises$classe<-as.character(weight_lifting_exercises$classe)  
weight_adaboost<-boosting(classe~., data = weight_lifting_exercises)
```

```
p.weight_adaboost<-predict(weight_adaboost,weight_lifting_exercises)  
head(p.weight_adaboost)  
head(p.weight_adaboost$class)  
p.weight_adaboost$confusion  
set.seed(300)  
car_adaboost_cv<-boosting.cv(classe,data = weight_lifting_exercises)  
car_adaboost_cv$confusion
```


Task 4:

1. Use the below-given data set

DataSet

2. Perform the below-given activities:
 - a. Create a classification model using different classifiers
 - b. Verify model goodness of fit
 - c. Apply all the model validation techniques.

#task 4 (session 16-20) assignment

#task 19 trial 1

#nn_classifiers

View(weight_lifting)

abcd <- weight_lifting

View(abcd)

dim(abcd)

str(abcd)

table(abcd\$user_name)

barplot(table(abcd\$user_name))

barplot(table(abcd\$user_name), col = c('green', 'red', 'blue', 'yellow', 'pink'),
main = 'Bar Plot of user name')

text(barplot(table(abcd\$user_name), col = c('green', 'red', 'blue', 'yellow', 'pink'),
main = 'Bar Plot of user name'), 0,
table(abcd\$user_name), cex = 2, pos = 3)

```
> #task 19 trial 1
> #nn_classifiers
> View(weight_lifting)
> abcd <- weight_lifting
> View(abcd)
> dim(abcd)
[1] 4024 159
> str(abcd)
```

```

Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    4024 obs. of  1
59 variables:
 $ user_name           : chr  "eurico" "eurico" "eurico" "eurico" ...
 $ raw_timestamp_part_1 : num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+
09 ...
 $ raw_timestamp_part_2 : num  34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp       : chr  "28/11/2011 14:15" "28/11/2011 14:15" "28/
11/2011 14:15" "28/11/2011 14:15" ...
 $ new_window           : chr  "no" "no" "no" "no" ...
 $ num_window           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ roll_belt            : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.3
1 2 ...
 $ pitch_belt           : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47
.4 47.7 ...
 $ yaw_belt             : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9
-82.2 -82.6 -82.8 ...
 $ total_accel_belt     : num  3 2 1 1 1 1 3 4 2 3 ...
 $ kurtosis_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ kurtosis_pitch_belt  : chr  NA NA NA NA ...
 $ kurtosis_yaw_belt    : chr  NA NA NA NA ...
 $ skewness_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ skewness_roll_belt_1 : chr  NA NA NA NA ...
 $ skewness_yaw_belt    : chr  NA NA NA NA ...
 $ max_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ max_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ max_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ min_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ min_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ min_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_yaw_belt   : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ avg_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_roll_belt     : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ var_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ avg_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_pitch_belt    : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ var_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ avg_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_yaw_belt      : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ var_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA NA ...
 $ gyros_belt_x         : num  2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.6
5 1.48 ...
 $ gyros_belt_y         : num  0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0
.03 -0.06 ...
 $ gyros_belt_z         : num  0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.
33 0.21 ...
 $ accel_belt_x         : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
 $ accel_belt_y         : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
 $ accel_belt_z         : num  22 16 8 7 0 -5 -8 -9 -7 1 ...
 $ magnet_belt_x        : num  387 405 409 422 418 432 438 440 443 449 ..
.
 $ magnet_belt_y        : num  525 512 511 513 508 510 508 503 507 499 ..
.

```

```

$ magnet_belt_z      : num  -267 -254 -244 -221 -208 -189 -176 -163 -1
40 -132 ...
$ roll_arm           : num   132 129 125 120 115 110 104 98.6 93.2 88.5
...
$ pitch_arm          : num   -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9
-49.7 -49 -48.1 ...
$ yaw_arm            : num   -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -1
1.4 -4.49 1.82 ...
$ total_accel_arm    : num    38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm    : num    NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm   : num    NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm     : num    NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x        : num    2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4
3.48 ...
$ gyros_arm_y        : num   -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83
-0.83 -0.83 -0.8 ...
$ gyros_arm_z        : num   -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -
0.21 -0.11 -0.15 ...
$ accel_arm_x        : num   143 146 156 158 163 160 165 153 143 135 ..
.
$ accel_arm_y        : num    30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z        : num   -346 -339 -307 -305 -288 -274 -225 -218 -2
05 -134 ...
$ magnet_arm_x       : num   556 599 613 646 670 696 721 725 740 741 ..
.
$ magnet_arm_y       : num   -205 -206 -198 -186 -175 -174 -161 -152 -1
33 -115 ...
$ magnet_arm_z       : num   -374 -335 -319 -268 -241 -193 -121 -105 -4
3 14 ...
$ kurtosis_roll_arm  : chr    NA NA NA NA ...
$ kurtosis_pitch_arm : chr    NA NA NA NA ...
$ kurtosis_yaw_arm   : chr    NA NA NA NA ...
$ skewness_roll_arm  : chr    NA NA NA NA ...
$ skewness_pitch_arm : chr    NA NA NA NA ...
$ skewness_yaw_arm   : chr    NA NA NA NA ...
$ max_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm       : num    NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm      : num    NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm        : num    NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm : num    NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm : num    NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm  : num    NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell      : num    51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell     : num    11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell       : num    104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num    NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_dumbbell : num    NA NA NA NA NA NA NA NA NA NA ...

```

```

$ kurtosis_yaw_dumbbell : chr NA NA NA NA ...
$ skewness_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell : chr NA NA NA NA ...
$ max_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell : num NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]
- attr(*, "spec")=List of 3
..$ cols :List of 159
.. ..$ user_name : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ raw_timestamp_part_1 : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ raw_timestamp_part_2 : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ cvtd_timestamp : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ new_window : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ num_window : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_belt_1 : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_belt : list()

```

```
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_total_accel_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_x : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_y : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_z : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_x : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_y : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_z : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_x : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_y : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_z : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_accel_arm : list()
```

```

.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_x : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_y : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_z : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_x : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_y : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_z : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_x : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_y : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_z : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_arm : list()

```

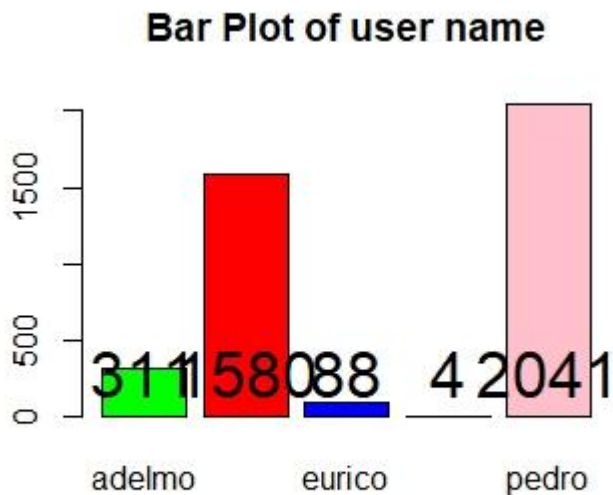
```

.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_arm : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. .. [list output truncated]
..$ default: list()
.. ..- attr(*, "class")= chr "collector_guess" "collector"
..$ skip : num 1
..- attr(*, "class")= chr "col_spec"
> table(abcd$user_name)

adelmo carlitos eurico jeremy pedro
311 1580 88 4 2041
> barplot(table(abcd$user_name))

```

```
> barplot(table(abcd$user_name), col = c('green', 'red', 'blue', 'yellow', 'pink'),
+         main = 'Bar Plot of user name')
> text(barplot(table(abcd$user_name), col = c('green', 'red', 'blue', 'yellow', 'pink'),
+         main = 'Bar Plot of user name'), 0,
+       table(abcd$user_name), cex = 2, pos = 3)
```



```
# recode diagnosis variable
abcd$user_name<- factor(abcd$user_name, levels = c('A', 'C', 'E', 'J', 'P'),
                        labels = c('Adelmo', 'Carlitos', 'Eurico', 'Jeremy', 'Pedro'))
table(abcd$user_name)
# replot the bar plot
text(barplot(table(abcd$user_name), col = c('green', 'red', 'blue', 'yellow', 'pink'),
                main = 'Bar Plot of user name'), 0,
      table(abcd$user_name), cex = 2, pos = 3)
# proportion
round(prop.table(table(abcd$user_name))*100, digits =2)

# normalization
normalize<- function(x){
  return((x-min(x))/(max(x)-min(x)))
}

normalize(c(1,2,3,4,5))
normalize(c(10,20,30,40,50))
```



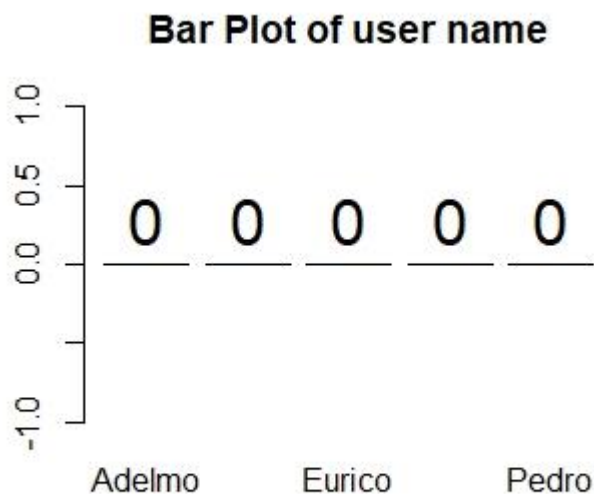
```

      Adelmo Carlitos      Eurico      Jeremy      Pedro
      0          0          0          0          0
> # replot the bar plot
> text(barplot(table(abcd$user_name), col = c('green', 'red', 'blue', 'yellow', 'pink'),
+       main = 'Bar Plot of user name'), 0,
+       table(abcd$user_name), cex = 2, pos = 3)
> # proportion
> round(prop.table(table(abcd$user_name))*100, digits =2)

      Adelmo Carlitos      Eurico      Jeremy      Pedro

> # normalization
> normalize<- function(x){
+   return((x-min(x))/(max(x)-min(x)))
+ }
> normalize(c(1,2,3,4,5))
[1] 0.00 0.25 0.50 0.75 1.00
> normalize(c(10,20,30,40,50))
[1] 0.00 0.25 0.50 0.75 1.00

```



```

# lapply() takes a list and applies a specified function
# to each item in the list
# start from 2 bcz 1st is a factor (diagnosis)
cvtd_timestamp<- as.numeric(abcd$ cvtd_timestamp)
new_window<- as.numeric(abcd$new_window)
kurtosis_picth_belt <- as.numeric(abcd$kurtosis_picth_belt)

```

```

kurtosis_yaw_belt <- as.numeric(abcd$kurtosis_yaw_belt)
skewness_roll_belt.1<- as.numeric(abcd$skewness_roll_belt.1)
skewness_yaw_belt<- as.numeric(abcd$skewness_yaw_belt)
kurtosis_roll_arm<- as.numeric(abcd$kurtosis_roll_arm)
kurtosis_pitch_arm<- as.numeric(abcd$kurtosis_pitch_arm)
kurtosis_yaw_arm<- as.numeric(abcd$kurtosis_yaw_arm)
skewness_roll_arm<- as.numeric(abcd$skewness_roll_arm)
skewness_pitch_arm<- as.numeric(abcd$skewness_pitch_arm)
skewness_yaw_arm<- as.numeric(abcd$skewness_yaw_arm)
kurtosis_yaw_dumbbell<- as.numeric(abcd$kurtosis_yaw_dumbbell)
skewness_yaw_dumbbell<- as.numeric(abcd$ skewness_yaw_dumbbell)
kurtosis_roll_forearm<- as.numeric(abcd$kurtosis_roll_forearm)
kurtosis_pitch_forearm<- as.numeric(abcd$kurtosis_pitch_forearm)
kurtosis_yaw_forearm<- as.numeric(abcd$kurtosis_yaw_forearm)
skewness_roll_forearm<- as.numeric(abcd$skewness_roll_forearm)
skewness_pitch_forearm<- as.numeric(abcd$skewness_pitch_forearm)
skewness_yaw_forearm<- as.numeric(abcd$skewness_yaw_forearm)
max_yaw_forearm<- as.numeric(abcd$max_yaw_forearm)
min_yaw_forearm<- as.numeric(abcd$min_yaw_forearm)
amplitude_yaw_forearm<- as.numeric(abcd$amplitude_yaw_forearm)
classe<- as.numeric(abcd$classe)

```

```

> cvtd_timestamp<- as.numeric(abcd$ cvtd_timestamp)
Warning message:
NAs introduced by coercion
> new_window<- as.numeric(abcd$new_window)
Warning message:
NAs introduced by coercion
> kurtosis_picth_belt <- as.numeric(abcd$kurtosis_picth_belt)
Warning message:
NAs introduced by coercion
> kurtosis_yaw_belt <- as.numeric(abcd$kurtosis_yaw_belt)
Warning message:
NAs introduced by coercion
> skewness_roll_belt.1<- as.numeric(abcd$skewness_roll_belt.1)
> skewness_yaw_belt<- as.numeric(abcd$skewness_yaw_belt)
Warning message:
NAs introduced by coercion
> kurtosis_roll_arm<- as.numeric(abcd$kurtosis_roll_arm)
Warning message:
NAs introduced by coercion
> kurtosis_pitch_arm<- as.numeric(abcd$kurtosis_pitch_arm)
> kurtosis_yaw_arm<- as.numeric(abcd$kurtosis_yaw_arm)
Warning message:
NAs introduced by coercion
> skewness_roll_arm<- as.numeric(abcd$skewness_roll_arm)
Warning message:
NAs introduced by coercion
> skewness_pitch_arm<- as.numeric(abcd$skewness_pitch_arm)
Warning message:

```

```

NAs introduced by coercion
> skewness_yaw_arm<- as.numeric(abcd$skewness_yaw_arm)
Warning message:
NAs introduced by coercion
> kurtosis_yaw_dumbbell<- as.numeric(abcd$kurtosis_yaw_dumbbell)
Warning message:
NAs introduced by coercion
> skewness_yaw_dumbbell<- as.numeric(abcd$ skewness_yaw_dumbbell)
Warning message:
NAs introduced by coercion
> kurtosis_roll_forearm<- as.numeric(abcd$kurtosis_roll_forearm)
Warning message:
NAs introduced by coercion
> kurtosis_pitch_forearm<- as.numeric(abcd$kurtosis_pitch_forearm)
> kurtosis_yaw_forearm<- as.numeric(abcd$kurtosis_yaw_forearm)
Warning message:
NAs introduced by coercion
> skewness_roll_forearm<- as.numeric(abcd$skewness_roll_forearm)
Warning message:
NAs introduced by coercion
> skewness_pitch_forearm<- as.numeric(abcd$skewness_pitch_forearm)
Warning message:
NAs introduced by coercion
> skewness_yaw_forearm<- as.numeric(abcd$skewness_yaw_forearm)
Warning message:
NAs introduced by coercion
> max_yaw_forearm<- as.numeric(abcd$max_yaw_forearm)
Warning message:
NAs introduced by coercion
> min_yaw_forearm<- as.numeric(abcd$min_yaw_forearm)
Warning message:
NAs introduced by coercion
> amplitude_yaw_forearm<- as.numeric(abcd$amplitude_yaw_forearm)
Warning message:
NAs introduced by coercion
> classe<- as.numeric(abcd$classe)
Warning message:
NAs introduced by coercion
> abcd_n<- as.data.frame(lapply(abcd[2:159], normalize))
Error in x - min(x) : non-numeric argument to binary operator
> wbcd_n<- as.data.frame(lapply(wbcd[2:31], normalize))

Error in lapply(wbcd[2:31], normalize) : object 'wbcd' not found
> str(abcd_n)
Error in str(abcd_n) : object 'abcd_n' not found
> str(abcd)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    4024 obs. of  1
59 variables:
 $ user_name           : Factor w/ 5 levels "Adelmo","Carlitos",...: NA N
A NA NA NA NA NA NA NA ...
 $ raw_timestamp_part_1 : num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+
09 ...
 $ raw_timestamp_part_2 : num  34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp       : chr   "28/11/2011 14:15" "28/11/2011 14:15" "28/
11/2011 14:15" "28/11/2011 14:15" ...
 $ new_window           : chr   "no" "no" "no" "no" ...

```

```

$ num_window      : num  1 1 1 1 1 1 1 1 1 1 ...
$ roll_belt       : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.3
1 2 ...
$ pitch_belt      : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47
.4 47.7 ...
$ yaw_belt        : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9
-82.2 -82.6 -82.8 ...
$ total_accel_belt : num  3 2 1 1 1 1 3 4 2 3 ...
$ kurtosis_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_belt : chr  NA NA NA NA ...
$ kurtosis_yaw_belt   : chr  NA NA NA NA ...
$ skewness_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
$ skewness_roll_belt_1 : chr  NA NA NA NA ...
$ skewness_yaw_belt   : chr  NA NA NA NA ...
$ max_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
$ gyros_belt_x        : num  2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.6
5 1.48 ...
$ gyros_belt_y        : num  0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0
.03 -0.06 ...
$ gyros_belt_z        : num  0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.
33 0.21 ...
$ accel_belt_x        : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y        : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z        : num  22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x       : num  387 405 409 422 418 432 438 440 443 449 ..
.
$ magnet_belt_y       : num  525 512 511 513 508 510 508 503 507 499 ..
.
$ magnet_belt_z       : num  -267 -254 -244 -221 -208 -189 -176 -163 -1
40 -132 ...
$ roll_arm           : num  132 129 125 120 115 110 104 98.6 93.2 88.5
...
$ pitch_arm          : num  -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9
-49.7 -49 -48.1 ...
$ yaw_arm            : num  -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -1
1.4 -4.49 1.82 ...
$ total_accel_arm     : num  38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm       : num  NA NA NA NA NA NA NA NA NA NA ...

```

```

$ avg_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm   : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x       : num  2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4
3.48 ...
$ gyros_arm_y       : num  -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83
-0.83 -0.83 -0.8 ...
$ gyros_arm_z       : num  -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -
0.21 -0.11 -0.15 ...
$ accel_arm_x       : num  143 146 156 158 163 160 165 153 143 135 ..
.
$ accel_arm_y       : num  30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z       : num  -346 -339 -307 -305 -288 -274 -225 -218 -2
05 -134 ...
$ magnet_arm_x      : num  556 599 613 646 670 696 721 725 740 741 ..
.
$ magnet_arm_y      : num  -205 -206 -198 -186 -175 -174 -161 -152 -1
33 -115 ...
$ magnet_arm_z      : num  -374 -335 -319 -268 -241 -193 -121 -105 -4
3 14 ...
$ kurtosis_roll_arm : chr  NA NA NA NA ...
$ kurtosis_pitch_arm : chr  NA NA NA NA ...
$ kurtosis_yaw_arm   : chr  NA NA NA NA ...
$ skewness_roll_arm  : chr  NA NA NA NA ...
$ skewness_pitch_arm : chr  NA NA NA NA ...
$ skewness_yaw_arm   : chr  NA NA NA NA ...
$ max_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm  : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell      : num  51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell     : num  11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell       : num  104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_yaw_dumbbell : chr  NA NA NA NA ...
$ skewness_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell : chr  NA NA NA NA ...
$ max_roll_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell  : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell   : num  NA NA NA NA NA NA NA NA NA NA ...

```

```

$ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell: num  NA NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]
- attr(*, "spec")=List of 3
..$ cols :List of 159
.. ..$ user_name : list()
.. ..$ raw_timestamp_part_1 : list()
.. ..$ raw_timestamp_part_2 : list()
.. ..$ cvtd_timestamp : list()
.. ..$ new_window : list()
.. ..$ num_window : list()
.. ..$ roll_belt : list()
.. ..$ pitch_belt : list()
.. ..$ yaw_belt : list()
.. ..$ total_accel_belt : list()
.. ..$ kurtosis_roll_belt : list()
.. ..$ kurtosis_pitch_belt : list()
.. ..$ kurtosis_yaw_belt : list()
.. ..$ skewness_roll_belt : list()
.. ..$ skewness_roll_belt_1 : list()
.. ..$ skewness_yaw_belt : list()
.. ..$ max_roll_belt : list()
.. ..$ max_pitch_belt : list()
.. ..$ max_yaw_belt : list()
.. ..$ min_roll_belt : list()
.. ..$ min_pitch_belt : list()
.. ..$ min_yaw_belt : list()
.. ..$ amplitude_roll_belt : list()
.. ..$ amplitude_pitch_belt : list()
.. ..$ amplitude_yaw_belt : list()
.. ..$ var_total_accel_belt : list()

```

```
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_belt      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_belt   : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_belt      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_belt     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_belt  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_belt     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_belt       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_belt    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_belt       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_x       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_y       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_z       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_x       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_y       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_z       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_x      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_arm           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_arm          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_arm            : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_accel_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_arm   : list()
```

```
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_x : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_y : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_z : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_x : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_y : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_z : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_x : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_y : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_z : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_arm : list()
```



```

.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. .. [list output truncated]
..$ default: list()
.. ..- attr(*, "class")= chr "collector_guess" "collector"
..$ skip : num 1
..- attr(*, "class")= chr "col_spec"

```

```
abcd_n<- as.data.frame(lapply(abcd[2:159], normalize))
```

```
wbcd_n<- as.data.frame(lapply(wbcd[2:31], normalize))
```

```
str(abcd_n)
```

```
str(abcd)
```

```
Factor w/ 5 levels "Adelmo","Carlitos",...: NA NA NA NA NA NA NA NA NA NA ..
```

```
# train and test data sets,
```

```
abcd_train<- abcd_n[1:3500, ]
```

```
abcd_test<- abcd_n[3501:4024, ]
```

```
# how we will compare the membership?
```

```
# label vector
abcd_train_labels<- abcd[1:3500, 1]
abcd_test_labels<- abcd[3501:4024, 1]
```

```
str(abcd_train)
str(abcd_train_labels)
```

```
str(abcd_test)
str(abcd_test_labels)
```

```
# training
install.packages('class')
library(class)
```

```
> abcd$user_name<- as.integer(abcd$user_name)
> str(abcd)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    4024 obs. of  1
59 variables:
 $ user_name           : int  NA NA NA NA NA NA NA NA NA NA ...
 $ raw_timestamp_part_1 : num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+
09 ...
 $ raw_timestamp_part_2 : num  34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp       : chr  "28/11/2011 14:15" "28/11/2011 14:15" "28/
11/2011 14:15" "28/11/2011 14:15" ...
 $ new_window           : chr  "no" "no" "no" "no" ...
 $ num_window           : num  1 1 1 1 1 1 1 1 1 1 ...
 $ roll_belt            : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.3
1 2 ...
 $ pitch_belt           : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47
.4 47.7 ...
 $ yaw_belt              : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9
-82.2 -82.6 -82.8 ...
 $ total_accel_belt     : num  3 2 1 1 1 1 3 4 2 3 ...
 $ kurtosis_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ kurtosis_pitch_belt  : chr  NA NA NA NA ...
 $ kurtosis_yaw_belt    : chr  NA NA NA NA ...
 $ skewness_roll_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ skewness_roll_belt_1 : chr  NA NA NA NA ...
 $ skewness_yaw_belt    : chr  NA NA NA NA ...
 $ max_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ max_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ max_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ min_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ min_pitch_belt       : num  NA NA NA NA NA NA NA NA NA NA ...
 $ min_yaw_belt         : num  NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_roll_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
 $ amplitude_yaw_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
 $ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA ...
 $ avg_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
 $ stddev_roll_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
 $ var_roll_belt        : num  NA NA NA NA NA NA NA NA NA NA ...
```

```

$ avg_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_belt      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_belt        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ gyros_belt_x        : num  2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.6
5 1.48 ...
$ gyros_belt_y        : num  0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0
.03 -0.06 ...
$ gyros_belt_z        : num  0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.
33 0.21 ...
$ accel_belt_x        : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y        : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z        : num  22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x       : num  387 405 409 422 418 432 438 440 443 449 ..
.
$ magnet_belt_y       : num  525 512 511 513 508 510 508 503 507 499 ..
.
$ magnet_belt_z       : num  -267 -254 -244 -221 -208 -189 -176 -163 -1
40 -132 ...
$ roll_arm            : num  132 129 125 120 115 110 104 98.6 93.2 88.5
...
$ pitch_arm           : num  -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9
-49.7 -49 -48.1 ...
$ yaw_arm             : num  -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -1
1.4 -4.49 1.82 ...
$ total_accel_arm     : num  38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm     : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm        : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm    : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm       : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm      : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm         : num  NA NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x         : num  2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4
3.48 ...
$ gyros_arm_y         : num  -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83
-0.83 -0.83 -0.8 ...
$ gyros_arm_z         : num  -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -
0.21 -0.11 -0.15 ...
$ accel_arm_x         : num  143 146 156 158 163 160 165 153 143 135 ..
.
$ accel_arm_y         : num  30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z         : num  -346 -339 -307 -305 -288 -274 -225 -218 -2
05 -134 ...
$ magnet_arm_x        : num  556 599 613 646 670 696 721 725 740 741 ..
.
$ magnet_arm_y        : num  -205 -206 -198 -186 -175 -174 -161 -152 -1
33 -115 ...
$ magnet_arm_z        : num  -374 -335 -319 -268 -241 -193 -121 -105 -4
3 14 ...
$ kurtosis_roll_arm   : chr  NA NA NA NA ...

```

```

$ kurtosis_picth_arm      : chr  NA NA NA NA ...
$ kurtosis_yaw_arm       : chr  NA NA NA NA ...
$ skewness_roll_arm      : chr  NA NA NA NA ...
$ skewness_pitch_arm     : chr  NA NA NA NA ...
$ skewness_yaw_arm       : chr  NA NA NA NA ...
$ max_roll_arm           : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ max_picth_arm          : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm            : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm           : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm          : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm            : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm     : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm    : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm      : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell          : num  51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell         : num  11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell           : num  104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_picth_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_yaw_dumbbell  : chr  NA NA NA NA ...
$ skewness_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell  : chr  NA NA NA NA ...
$ max_roll_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ max_picth_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell      : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell     : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell       : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell : num  NA NA NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]
- attr(*, "spec")=List of 3
..$ cols      :List of 159
.. ..$ user_name      : list()
.. ..$- attr(*, "class")= chr  "collector_character" "collector"
.. ..$ raw_timestamp_part_1 : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ raw_timestamp_part_2 : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ cvtd_timestamp    : list()
.. ..$- attr(*, "class")= chr  "collector_character" "collector"
.. ..$ new_window       : list()
.. ..$- attr(*, "class")= chr  "collector_character" "collector"
.. ..$ num_window       : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ roll_belt        : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ pitch_belt       : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ yaw_belt         : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ total_accel_belt  : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ kurtosis_roll_belt : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"

```

```
.. ..$ kurtosis_pitch_belt      : list()
.. ..- attr(*, "class")= chr   "collector_character" "collector"
.. ..$ kurtosis_yaw_belt        : list()
.. ..- attr(*, "class")= chr   "collector_character" "collector"
.. ..$ skewness_roll_belt       : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ skewness_roll_belt_1     : list()
.. ..- attr(*, "class")= chr   "collector_character" "collector"
.. ..$ skewness_yaw_belt        : list()
.. ..- attr(*, "class")= chr   "collector_character" "collector"
.. ..$ max_roll_belt            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ max_pitch_belt           : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ max_yaw_belt             : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ min_roll_belt           : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ min_pitch_belt          : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ min_yaw_belt            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ amplitude_roll_belt      : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ amplitude_pitch_belt     : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ amplitude_yaw_belt       : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ var_total_accel_belt     : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ avg_roll_belt            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ stddev_roll_belt         : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ var_roll_belt           : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ avg_pitch_belt          : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ stddev_pitch_belt       : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ var_pitch_belt          : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ avg_yaw_belt            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ stddev_yaw_belt         : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ var_yaw_belt            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ gyros_belt_x             : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ gyros_belt_y            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ gyros_belt_z            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ accel_belt_x            : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
```

```
.. ..$ accel_belt_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_x      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_arm           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_arm          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_arm            : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_accel_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_arm   : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_arm        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_arm        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_x        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_y        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_z        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_x        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_y        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_z        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_x       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_y       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_z       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
```

```

.. ..$ kurtosis_roll_arm      : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_pitch_arm     : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_yaw_arm       : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_arm      : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_pitch_arm     : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_yaw_arm       : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_arm           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_arm          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_arm            : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_arm           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_arm          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_arm            : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_dumbbell          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_dumbbell         : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_dumbbell           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_dumbbell  : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_dumbbell  : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_dumbbell      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_dumbbell     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_dumbbell       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_dumbbell      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"

```

```

.. ..$ min_pitch_dumbbell      : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ min_yaw_dumbbell        : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ amplitude_roll_dumbbell : list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. ..$ amplitude_pitch_dumbbell: list()
.. ..- attr(*, "class")= chr   "collector_double" "collector"
.. .. [list output truncated]
..$ default: list()
.. ..- attr(*, "class")= chr   "collector_guess" "collector"
..$ skip      : num 1
.. ..- attr(*, "class")= chr   "col_spec"
> normalize<- function(x) {
+   return((x-min(x))/(max(x)-min(x)))
+ }
> normalize(c(1,2,3,4,5))
[1] 0.00 0.25 0.50 0.75 1.00
> normalize(c(10,20,30,40,50))
[1] 0.00 0.25 0.50 0.75 1.00

```

```

# wow! training and testing simultaneously!
# choose k as approx sqrt of n in training data set
# sqrt 469 = approx 21

```

```

abcd_test_pred<- knn(train = abcd_train, test = abcd_test,
                     cl = abcd_train_labels, k= 61)

```

```

summary(abcd_test_pred)

```

```

# evaluate model performance

```

```

install.packages('gmodels')
library(gmodels)

```

```

(CrossTable(x = abcd_test_labels, y = abcd_test_pred,
            prop.chisq = FALSE))

```

```

#ann classification
# before normalizing, diagnosis has to be an integer (or numeric)
abcd$user_name<- as.integer(abcd$user_name)
str(abcd)

```

```

normalize<- function(x) {
  return((x-min(x))/(max(x)-min(x)))
}

```



```
normalize(c(1,2,3,4,5))
normalize(c(10,20,30,40,50))
```

```
abcd_n<- as.data.frame(lapply(abcd[2:159], normalize))
# id is not normalised
str(abcd_n)
train_ann<- abcd_n[1:3500,]
test_ann<- wbcd_n[3501:4024,]
```

```
str(train_ann)
```

```
library(neuralnet)
```

```
ann_train<-
neuralnet(user_name~raw_timestamp_part_1+raw_timestamp_part_2+cvtd_timestamp+new_
window+num_window+roll_belt+pitch_belt+yaw_belt+
```

```
total_accel_belt+gyros_belt_x+gyros_belt_y+gyros_belt_z+accel_belt_x+accel_belt_y+accel_
belt_z+magnet_belt_x+magnet_belt_y+magnet_belt_z+roll_arm+
```

```
pitch_arm+yaw_arm+total_accel_arm+gyros_arm_x+gyros_arm_y+gyros_arm_z+accel_arm_
x+accel_arm_y+accel_arm_z+magnet_arm_x+magnet_arm_y+magnet_arm_z+
```

```
roll_dumbbell+pitch_dumbbell+yaw_dumbbell+total_accel_dumbbell+gyros_dumbbell_x+gyro
s_dumbbell_y+gyros_dumbbell_z+accel_dumbbell_x+accel_dumbbell_y+
```

```
accel_dumbbell_z+magnet_dumbbell_x+magnet_dumbbell_y+magnet_dumbbell_z+roll_forea
rm+pitch_forearm+yaw_forearm+total_accel_forearm+gyros_forearm_x+
```

```
gyros_forearm_y+gyros_forearm_z+accel_forearm_x+accel_forearm_y+accel_forearm_z+ma
gnet_forearm_x+magnet_forearm_y+magnet_forearm_z+classe,
      data = train_ann,
      hidden = c(2,1),
      linear.output = FALSE,
      threshold = 0.01)
```

```
ann_train$result.matrix
```

```
plot(ann_train)
```

```
str(test_ann)
test.ann<- subset(test_ann[,-1])
str(test.ann)
```

```

ann.results<- compute(ann_train, test.ann)
head(ann.results$net.result, 5)

# Accuracy
results<- data.frame(actual = test_ann$user_name,
                      prediction = ann.results$net.result)

roundedresults<- sapply(results, round, digits = 0)

roundedresults$df<- data.frame(roundedresults)

attach(roundedresults$df)

table(actual, prediction)

dim(test_ann)

# letters classifier
abcd <- read.csv("C:/Users/satish/Desktop/abcd.csv")
View(abcd)
letters<- abcd
dim(letters)
str(letters)
names(letters)

```

```

> dim(letters)
[1] 4024 159
> str(letters)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 4024 obs. of 159 variables:
 $ user_name      : int  NA NA NA NA NA NA NA NA NA NA NA ...
 $ raw_timestamp_part_1 : num  1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+09 ...
 $ raw_timestamp_part_2 : num  34670 62641 70653 82654 90637 ...
 $ cvtd_timestamp   : chr   "28/11/2011 14:15" "28/11/2011 14:15" "28/11/2011 14:15" ...
 $ new_window       : chr   "no" "no" "no" "no" ...
 $ num_window        : num    1 1 1 1 1 1 1 1 1 1 ...
 $ roll_belt         : num    3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.31 2 ...
 $ pitch_belt        : num    41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47.4 47.7 ...
 $ yaw_belt          : num   -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9 -82.2 -82.6 -82.8 ...
 $ total_accel_belt   : num    3 2 1 1 1 1 3 4 2 3 ...
 $ kurtosis_roll_belt : num   NA NA NA NA NA NA NA NA NA NA ...
 $ kurtosis_pitch_belt : chr   NA NA NA NA ...
 $ kurtosis_yaw_belt  : chr   NA NA NA NA ...
 $ skewness_roll_belt : num   NA NA NA NA NA NA NA NA NA NA ...

```

```

$ skewness_roll_belt_1 : chr NA NA NA NA ...
$ skewness_yaw_belt : chr NA NA NA NA ...
$ max_roll_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_total_accel_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_belt : num NA NA NA NA NA NA NA NA NA NA NA ...
$ gyros_belt_x : num 2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.6
5 1.48 ...
$ gyros_belt_y : num 0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0
.03 -0.06 ...
$ gyros_belt_z : num 0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.
33 0.21 ...
$ accel_belt_x : num -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y : num -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z : num 22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x : num 387 405 409 422 418 432 438 440 443 449 ..
.
$ magnet_belt_y : num 525 512 511 513 508 510 508 503 507 499 ..
.
$ magnet_belt_z : num -267 -254 -244 -221 -208 -189 -176 -163 -1
40 -132 ...
$ roll_arm : num 132 129 125 120 115 110 104 98.6 93.2 88.5
...
$ pitch_arm : num -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9
-49.7 -49 -48.1 ...
$ yaw_arm : num -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -1
1.4 -4.49 1.82 ...
$ total_accel_arm : num 38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm : num NA NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x : num 2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4
3.48 ...

```

```

$ gyros_arm_y      : num  -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83
-0.83 -0.83 -0.8 ...
$ gyros_arm_z      : num  -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -
0.21 -0.11 -0.15 ...
$ accel_arm_x      : num   143 146 156 158 163 160 165 153 143 135 ..
.
$ accel_arm_y      : num   30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z      : num  -346 -339 -307 -305 -288 -274 -225 -218 -2
05 -134 ...
$ magnet_arm_x     : num   556 599 613 646 670 696 721 725 740 741 ..
.
$ magnet_arm_y     : num  -205 -206 -198 -186 -175 -174 -161 -152 -1
33 -115 ...
$ magnet_arm_z     : num  -374 -335 -319 -268 -241 -193 -121 -105 -4
3 14 ...
$ kurtosis_roll_arm : chr   NA NA NA NA ...
$ kurtosis_pitch_arm : chr   NA NA NA NA ...
$ kurtosis_yaw_arm   : chr   NA NA NA NA ...
$ skewness_roll_arm  : chr   NA NA NA NA ...
$ skewness_pitch_arm : chr   NA NA NA NA ...
$ skewness_yaw_arm   : chr   NA NA NA NA ...
$ max_roll_arm       : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_arm      : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm        : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm       : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm      : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm        : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm  : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell      : num   51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell     : num   11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell       : num  104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_yaw_dumbbell : chr   NA NA NA NA ...
$ skewness_roll_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell : chr   NA NA NA NA ...
$ max_roll_dumbbell  : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell   : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell  : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell   : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell : num   NA NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]
- attr(*, "spec")=List of 3
..$ cols :List of 159
.. ..$ user_name      : list()
.. ..$- attr(*, "class")= chr  "collector_character" "collector"
.. ..$ raw_timestamp_part_1 : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ raw_timestamp_part_2 : list()
.. ..$- attr(*, "class")= chr  "collector_double" "collector"

```

```
.. ..$ cvtd_timestamp      : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ new_window          : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ num_window          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_belt           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_belt          : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_belt            : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_belt    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_belt  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_yaw_belt   : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_belt  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_belt_1 : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_yaw_belt   : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_belt       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_belt      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_belt        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_belt       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_belt      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_belt        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_belt  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_total_accel_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_belt       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_belt    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_belt       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_belt      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_belt   : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
```

```
.. ..$ var_pitch_belt      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_belt        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_belt     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_belt        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_x        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_y        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_belt_z        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_x        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_y        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_belt_z        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_x       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_y       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_belt_z       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_arm            : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_arm           : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_arm             : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_accel_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_arm        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_arm        : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_arm         : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_yaw_arm         : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_x         : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
```

```

.. ..$ gyros_arm_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ gyros_arm_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_x      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_x     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_y     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_z     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ kurtosis_yaw_arm  : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_yaw_arm  : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_arm    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_dumbbell    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_dumbbell   : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_dumbbell     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"

```

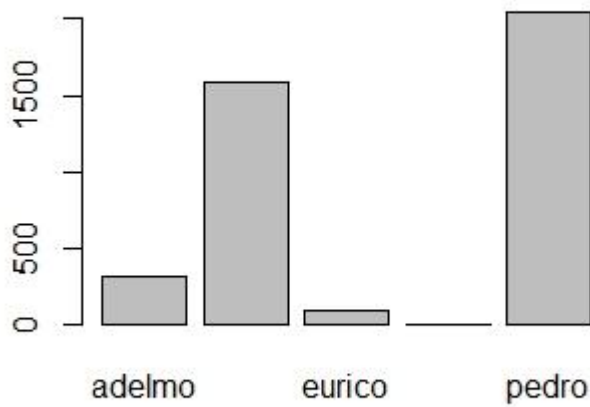
```

.. ..$ kurtosis_yaw_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ skewness_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. .. [list output truncated]
..$ default: list()
.. ..- attr(*, "class")= chr "collector_guess" "collector"
..$ skip : num 1
..- attr(*, "class")= chr "col_spec"
> names(letters)
[1] "user_name" "raw_timestamp_part_1" "raw_timestamp_p
art_2"
[4] "cvtd_timestamp" "new_window" "num_window"
[7] "roll_belt" "pitch_belt" "yaw_belt"
[10] "total_accel_belt" "kurtosis_roll_belt" "kurtosis_pitch_
belt"
[13] "kurtosis_yaw_belt" "skewness_roll_belt" "skewness_roll_b
elt_1"
[16] "skewness_yaw_belt" "max_roll_belt" "max_pitch_belt"
[19] "max_yaw_belt" "min_roll_belt" "min_pitch_belt"
[22] "min_yaw_belt" "amplitude_roll_belt" "amplitude_pitch
belt"
[25] "amplitude_yaw_belt" "var_total_accel_belt" "avg_roll_belt"
[28] "stddev_roll_belt" "var_roll_belt" "avg_pitch_belt"
[31] "stddev_pitch_belt" "var_pitch_belt" "avg_yaw_belt"
[34] "stddev_yaw_belt" "var_yaw_belt" "gyros_belt_x"
[37] "gyros_belt_y" "gyros_belt_z" "accel_belt_x"

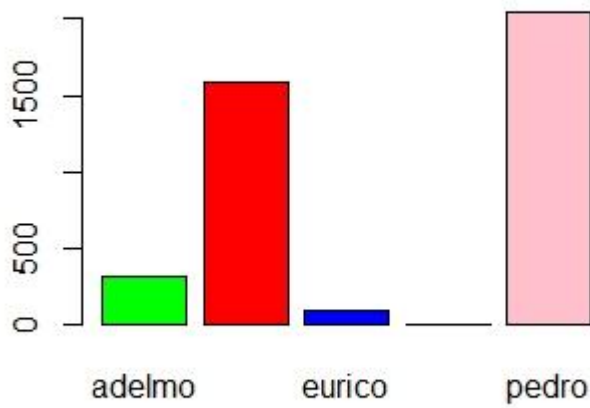
```


[40]	"accel_belt_y"	"accel_belt_z"	"magnet_belt_x"
[43]	"magnet_belt_y"	"magnet_belt_z"	"roll_arm"
[46]	"pitch_arm"	"yaw_arm"	"total_accel_arm"
[49]	"var_accel_arm"	"avg_roll_arm"	"stddev_roll_arm"
[52]	"var_roll_arm"	"avg_pitch_arm"	"stddev_pitch_arm"
[55]	"var_pitch_arm"	"avg_yaw_arm"	"stddev_yaw_arm"
[58]	"var_yaw_arm"	"gyros_arm_x"	"gyros_arm_y"
[61]	"gyros_arm_z"	"accel_arm_x"	"accel_arm_y"
[64]	"accel_arm_z"	"magnet_arm_x"	"magnet_arm_y"
[67]	"magnet_arm_z"	"kurtosis_roll_arm"	"kurtosis_pitch_arm"
[70]	"kurtosis_yaw_arm"	"skewness_roll_arm"	"skewness_pitch_arm"
[73]	"skewness_yaw_arm"	"max_roll_arm"	"max_pitch_arm"
[76]	"max_yaw_arm"	"min_roll_arm"	"min_pitch_arm"
[79]	"min_yaw_arm"	"amplitude_roll_arm"	"amplitude_pitch_arm"
[82]	"amplitude_yaw_arm"	"roll_dumbbell"	"pitch_dumbbell"
[85]	"yaw_dumbbell"	"kurtosis_roll_dumbbell"	"kurtosis_pitch_dumbbell"
[88]	"kurtosis_yaw_dumbbell"	"skewness_roll_dumbbell"	"skewness_pitch_dumbbell"
[91]	"skewness_yaw_dumbbell"	"max_roll_dumbbell"	"max_pitch_dumbbell"
[94]	"max_yaw_dumbbell"	"min_roll_dumbbell"	"min_pitch_dumbbell"
[97]	"min_yaw_dumbbell"	"amplitude_roll_dumbbell"	"amplitude_pitch_dumbbell"
[100]	"amplitude_yaw_dumbbell"	"total_accel_dumbbell"	"var_accel_dumbbell"
[103]	"avg_roll_dumbbell"	"stddev_roll_dumbbell"	"var_roll_dumbbell"
[106]	"avg_pitch_dumbbell"	"stddev_pitch_dumbbell"	"var_pitch_dumbbell"
[109]	"avg_yaw_dumbbell"	"stddev_yaw_dumbbell"	"var_yaw_dumbbell"
[112]	"gyros_dumbbell_x"	"gyros_dumbbell_y"	"gyros_dumbbell_z"
[115]	"accel_dumbbell_x"	"accel_dumbbell_y"	"accel_dumbbell_z"
[118]	"magnet_dumbbell_x"	"magnet_dumbbell_y"	"magnet_dumbbell_z"
[121]	"roll_forearm"	"pitch_forearm"	"yaw_forearm"

[124]	"kurtosis_roll_forearm"	"kurtosis_picth_forearm"	"kurtosis_yaw_fo
	rearm"		
[127]	"skewness_roll_forearm"	"skewness_pitch_forearm"	"skewness_yaw_fo
	rearm"		
[130]	"max_roll_forearm"	"max_picth_forearm"	"max_yaw_forearm
	"		
[133]	"min_roll_forearm"	"min_pitch_forearm"	"min_yaw_forearm
	"		
[136]	"amplitude_roll_forearm"	"amplitude_pitch_forearm"	"amplitude_yaw_f
	orearm"		
[139]	"total_accel_forearm"	"var_accel_forearm"	"avg_roll_forear
	m"		
[142]	"stddev_roll_forearm"	"var_roll_forearm"	"avg_pitch_forea
	rm"		
[145]	"stddev_pitch_forearm"	"var_pitch_forearm"	"avg_yaw_forearm
	"		
[148]	"stddev_yaw_forearm"	"var_yaw_forearm"	"gyros_forearm_x
	"		
[151]	"gyros_forearm_y"	"gyros_forearm_z"	"accel_forearm_x
	"		
[154]	"accel_forearm_y"	"accel_forearm_z"	"magnet_forearm_
	x"		
[157]	"magnet_forearm_y"	"magnet_forearm_z"	"classe"



Bar Plot of user name



```
letters_train<- letters[1:3500, ]
letters_test<- letters[3501:4024, ]
```

```
install.packages('kernlab')
library(kernlab)
```

```
letter_classifier<- ksvm(user_name~., data = letters_train,
                          kernel = 'vanilladot')
```

```
letter_classifier

#test
letter_predictions<- predict(letter_classifier, letters_test)

head(letter_predictions)

table(letter_predictions, letters_test$letter)

agreement<- letter_predictions == letters_test$letter

table(agreement)
prop.table(table(agreement))

#____improving model
set.seed(12345)
letter_classifier_rbf<- ksvm(user_name~., data = letters_train,
                             kernel = 'rbfdot')
# it takes 1 to 3 minutes, mine is 16 GB RAM

letter_predictions_rbf<- predict(letter_classifier_rbf,
                                 letters_test)

agreement_rbf<- letter_predictions_rbf == letters_test$letter

table(agreement_rbf)
prop.table(table(agreement_rbf))
```

Task 5:

1. Use the below given data set

Data Set

2. Perform the below given activities:

a. Create classification model using different random forest models

b. Verify model goodness of fit

c. Apply all the model validation techniques

d. Make conclusions

e. Plot importance of variables

```
wle<- Example_WearableComputing_weight_lifting_exercises_biceps_curl_variations_task17
wle<-wle[,-c(2:6,11:35,49:58,68:82,86:100,102:111,124:138,140:149)]
library(psych)
t<-View(describe(wle))
sum(is.na(wle))
summary(wle)
dim(wle)
```

```
> sum(is.na(wle))
[1] 108
> summary(wle)
EXAMPLE DATA: get the complete file at http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises
Length:4027

Class :character

Mode :character

      X7              X8              X9              X10
X36
Length:4027      Length:4027      Length:4027      Length:4027
Length:4027
Class :character  Class :character  Class :character  Class :character
Class :character
```

Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X37	X38	X39	X40
X41			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			
Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X42	X43	X44	X45
X46			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			
Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X47	X48	X59	X60
X61			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			
Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X62	X63	X64	X65
X66			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			
Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X67	X83	X84	X85
X101			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			
Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X112	X113	X114	X115
X116			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			
Mode :character	Mode :character	Mode :character	Mode :character
Mode :character			
X117	X118	X119	X120
X121			
Length:4027	Length:4027	Length:4027	Length:4027
Length:4027			
Class :character	Class :character	Class :character	Class :character
Class :character			

```

Mode :character Mode :character Mode :character Mode :character
Mode :character
X122 X123 X139 X150
X151
Length:4027 Length:4027 Length:4027 Length:4027
Length:4027
Class :character Class :character Class :character Class :character
Class :character
Mode :character Mode :character Mode :character Mode :character
Mode :character
X152 X153 X154 X155
X156
Length:4027 Length:4027 Length:4027 Length:4027
Length:4027
Class :character Class :character Class :character Class :character
Class :character
Mode :character Mode :character Mode :character Mode :character
Mode :character
X157 X158 X159
Length:4027 Length:4027 Length:4027
Class :character Class :character Class :character
Mode :character Mode :character Mode :character
> dim(wle)
[1] 4027 54
> pairs(wle[,1:10])

```

```
pairs(wle[,1:10])
```

```
library(ISLR)
```

smp_siz<-floor(0.75*nrow(wle)) # creates a value for dividing the data into train and test. In this case the value is defined as 75% of the number of rows in the dataset

```
smp_siz
```

```
set.seed(123) # set seed to ensure you always have same random numbers generated
```

```
train_ind = sample(seq_len(nrow(wle)),size = smp_siz) # Randomly identifies the rows equal to sample size ( defined in previous instruction) from all the rows of Smarket dataset and stores the row number in train_ind
```

```
wle_train =wle[train_ind,] #creates the training dataset with row numbers stored in train_ind
```

```
wle_test=wle[-train_ind,]
```

```

> smp_siz<-floor(0.75*nrow(wle)) # creates a value for dividing the data in
to train and test. In this case the value is defined as 75% of the number of
rows in the dataset
> smp_siz
[1] 3020

```

```
#### using tree model
```

```
library(tree)
```

```
fit1<-tree(classe~., data = wle)
```

```
plot(fit1)
```

```
text(fit1)
summary(fit1)
pred<-predict(fit1, wle, type="class")
confmat<-confusionMatrix(pred, wle$classe)
confmat
```

#1. create a classification model using different classifiers

```
library(caret)
library(rpart)
```

##using cv

```
train_control<- trainControl(method="cv", number=3)
```

```
model<- train(classe ~., data=wle_train,trControl=train_control, method="rf")
```

```
model
```

```
predictions<- predict(model,wle_test)
```

```
pred<- cbind(wle_test,predictions)
```

```
confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)
```

```
confusionMatrix
```

##using repeatedcv

```
control <- trainControl(method = 'repeatedcv',
                        number = 3,
                        repeats = 3)
```

```
set.seed(123)
```

```
mtry <- sqrt(ncol(wle_train))
```

```
tuneGrid <- expand.grid(.mtry=mtry)
```

```
rf_default <- train(classe~.,data = wle_train,method = 'rf',tuneGrid = tuneGrid,trControl =
control)
```

```
print(rf_default)
```

```
predictions<- predict(rf_default,wle_test)
```

```
pred<- cbind(wle_test,predictions)
```

```
confusionMatrix<- confusionMatrix(pred$predictions, pred$classe)
```

```
varImp(rf_default)
```

```
plot(varImp(rf_default))
```

###classification model using different classifiers

```
control <- trainControl(method = 'repeatedcv',
                        number = 3,
```



```

        repeats = 3)
metric<-'Accuracy'
preProcess=c("center", "scale")

# Linear Discriminant Analysis
set.seed(7)
fit.lda <- train(classe~.,data = wle_train, method="lda", metric=metric, preProc=c("center",
"scale"), trControl=control)
# Logistic Regression
set.seed(7)
fit.glm <- train(classe~.,data = wle_train, method="glm", metric=metric, trControl=control)
# GLMNET
set.seed(7)
fit.glmnet <- train(classe~.,data = wle_train, method="glmnet", metric=metric,
preProc=c("center", "scale"), trControl=control)
# SVM Radial
set.seed(7)
fit.svmRadial <- train(classe~.,data = wle_train, method="svmRadial", metric=metric,
preProc=c("center", "scale"), trControl=control, fit=FALSE)
# kNN
set.seed(7)
fit.knn <- train(classe~.,data = wle_train, method="knn", metric=metric, preProc=c("center",
"scale"), trControl=control)
# Naive Bayes
set.seed(7)
fit.nb <- train(classe~.,data = wle_train, method="nb", metric=metric, trControl=control)
# CART
set.seed(7)
fit.cart <- train(classe~.,data = wle_train, method="rpart", metric=metric, trControl=control)
# C5.0
set.seed(7)
fit.c50 <- train(classe~.,data = wle_train, method="C5.0", metric=metric, trControl=control)
# Bagged CART
set.seed(7)
fit.treebag <- train(classe~.,data = wle_train, method="treebag", metric=metric,
trControl=control)
# Random Forest
set.seed(7)
fit.rf <- train(classe~.,data = wle_train, method="rf", metric=metric, trControl=control)
# Stochastic Gradient Boosting (Generalized Boosted Modeling)
set.seed(7)
fit.gbm <- train(classe~.,data = wle_train, method="gbm", metric=metric, trControl=control,
verbose=FALSE)

```

```
> model
Random Forest

3018 samples
  53 predictor
   5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (3 fold)
Summary of sample sizes: 2012, 2012, 2012
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2    0.9973492 0.9962935
   29    0.9960239 0.9944440
   56    0.9943671 0.9921247

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

classifiers results comparision

```
results <- resamples(list(lda=fit.lda,glmnet=fit.glmnet,
                        svm=fit.svmRadial, knn=fit.knn, nb=fit.nb, cart=fit.cart, c50=fit.c50,
                        bagging=fit.treebag, rf=fit.rf, gbm=fit.gbm))
# Table comparison
summary(results)
```

```
print(rf_default)
Random Forest

3018 samples
  53 predictor
   5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing
Resampling: Cross-Validated (3 fold, repeated 3 times)
Summary of sample sizes: 2012, 2011, 2013, 2012, 2013, 2011, ...
Resampling results:

  Accuracy   Kappa
0.9980124 0.9972211

Tuning parameter 'mtry' was held constant at a value of 7.348469
```

```
# boxplot comparison
bwplot(results)
# Dot-plot comparison
dotplot(results)
```

#2.verify model goodness of fit

```
library(C50)
fitt<-glm(classe~gyros_forearm_x,data = wle_train, family = binomial(link = 'logit'))
fitt
```

```
library(ResourceSelection)
tes<-hoslem.test(fitt$y,fitted(fitt))
```

```
cbind(tes$observed,tes$expected)
```

```
#plot the fitted model
plot(fitt$fitted.values)
```

```
x <- wle_train$classe
y <- wle_train$total_accel_belt
library(MASS)
boxcox(y~x)
plot(1/y^2~x)
```

```
#check for multicollinearity
library(car)
vif(fitt)
vif(step_fit)
```

```
pred <- predict(fitt,newdata = wle_test,type = 'response')
```

```
#check the AUC curve
library(pROC)
g <- roc(classe~gyros_forearm_x, data = wle_test)
g
plot(g)
```

#3.Apply all the model validation techniques

```
library(caret)
library(rpart)
```

```
control <- trainControl(method = 'repeatedcv',
                        number = 5,
                        repeats = 3,
                        search = 'grid')
```

```

library(C50)
set.seed(123)
metric <- 'Accuracy'
gbm_mod <- train(user_name~.,
                 data = wle_train,
                 method = 'gbm',
                 metric = metric,
                 trControl = control)
print(gbm_mod)
plot(gbm_mod)

summary(gbm_mod)

predictions<- predict(gbm_mod,wle_test)

pred<- cbind(wle_test,predictions)

# summarize results
confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)
confusionMatrix

#####Grid search

control <- trainControl(method = 'repeatedcv',
                        number = 5,
                        repeats = 3,
                        search = 'grid')
set.seed(123)
tunegrid <- expand.grid(.mtry=c(1:80))
mtry <- sqrt(ncol(x))
rf_gridsearch <- train(classe~.,
                      data = wle_train[1:200,],
                      method = 'rf',
                      tuneGrid = tunegrid,
                      trControl = control)
print(rf_gridsearch)
plot(rf_gridsearch)

```

```
> print(rf_gridsearch)
```

Random Forest

200 samples

53 predictor

5 classes: 'A', 'B', 'C', 'D', 'E'

No pre-processing

Resampling: Cross-Validated (5 fold, repeated 3 times)

Summary of sample sizes: 161, 160, 159, 160, 160, 160, ...

Resampling results across tuning parameters:

mtry	Accuracy	Kappa
1	0.9030736	0.8535643
2	0.9216187	0.8830259
3	0.9316759	0.8988623
4	0.9249154	0.8882901
5	0.9283791	0.8936997
6	0.9267081	0.8912990
7	0.9234585	0.8864467
8	0.9250458	0.8890748
9	0.9232060	0.8861876
10	0.9200781	0.8818594
11	0.9184992	0.8794023
12	0.9184564	0.8797526
13	0.9200415	0.8815916
14	0.9201681	0.8821932
15	0.9150865	0.8747708
16	0.9083604	0.8644985
17	0.9151231	0.8750222
18	0.9133237	0.8718563
19	0.9118691	0.8702866
20	0.9135785	0.8725135
21	0.9083687	0.8649802
22	0.9102025	0.8677069
23	0.9101147	0.8674175
24	0.9050270	0.8598105
25	0.9117898	0.8702253
26	0.9033604	0.8571382
27	0.9067898	0.8627541
28	0.9085358	0.8655443
29	0.9083687	0.8651572
30	0.9084971	0.8653777
31	0.9084971	0.8655501
32	0.9035764	0.8582878
33	0.9068221	0.8633137
34	0.9051554	0.8606386
35	0.9100760	0.8676206
36	0.9002004	0.8533217
37	0.9069098	0.8633606
38	0.9001210	0.8529106
39	0.9051637	0.8604547
40	0.9051637	0.8604184
41	0.9017793	0.8557320
42	0.9050760	0.8603996
43	0.9052431	0.8610768
44	0.9051210	0.8605718

```

45 0.9035764 0.8584229
46 0.9001920 0.8533574
47 0.9034971 0.8577139
48 0.9034094 0.8579418
49 0.9034971 0.8580458
50 0.9050760 0.8602990
51 0.9001127 0.8527202
52 0.9002797 0.8539340
53 0.9067877 0.8631419
54 0.9034543 0.8583463
55 0.9034971 0.8579276
56 0.8952348 0.8455947
57 0.9001981 0.8528013
58 0.8986537 0.8510801
59 0.9084971 0.8655240
60 0.9051637 0.8603684
61 0.9051637 0.8604126
62 0.9002004 0.8531701
63 0.8984971 0.8505917
64 0.9002004 0.8529107
65 0.8950333 0.8453989
66 0.9017877 0.8553699
67 0.8969077 0.8479920
68 0.9002838 0.8533895
69 0.9019098 0.8561315
70 0.9051637 0.8604977
71 0.8985681 0.8509688
72 0.9003631 0.8538486
73 0.9034521 0.8577027
74 0.9068731 0.8633277
75 0.8917813 0.8402895
76 0.9001210 0.8531087
77 0.9034971 0.8579992
78 0.9002858 0.8530880
79 0.8969442 0.8482797
80 0.8936558 0.8431939

```

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 3.

```
predictions<- predict(rf_gridsearch,wle_test)
```

```
pred<- cbind(wle_test,predictions)
```

```
# summarize results
```

```
library(caret)
```

```
confusionMatrix<- confusionMatrix(pred$predictions,pred$classe)
confusionMatrix
```

```
> confusionMatrix
function (data, ...)
```

```

{
  UseMethod("confusionMatrix")
}
<bytecode: 0x0000027683564018>
<environment: namespace:caret>
> varImp(rf_gridsearch)
rf variable importance

  only 20 most important variables shown (out of 55)

              Overall
pitch_belt      100.00
magnet_dumbbell_y 98.23
roll_dumbbell   79.77
magnet_dumbbell_x 76.82
accel_belt_x    74.87
magnet_dumbbell_z 73.96
gyros_belt_x    72.48
yaw_belt        71.88
accel_belt_y    70.22
accel_dumbbell_y 68.79
user_namepedro  67.42
magnet_belt_x   65.96
yaw_dumbbell    63.27
gyros_belt_z    54.78
roll_belt       53.77
accel_belt_z    50.25
accel_dumbbell_z 50.08
accel_forearm_x 48.35
total_accel_dumbbell 46.08
pitch_forearm   45.82

```

```
varImp(rf_gridsearch)
```

#d. Make conclusions

classifiers results comparision

```

results <- resamples(list(lda=fit.lda,glmnet=fit.glmnet,
                        svm=fit.svmRadial, knn=fit.knn, nb=fit.nb, cart=fit.cart, c50=fit.c50,
                        bagging=fit.treebag, rf=fit.rf, gbm=fit.gbm))

```

Table comparison

```
summary(results)
```

boxplot comparison

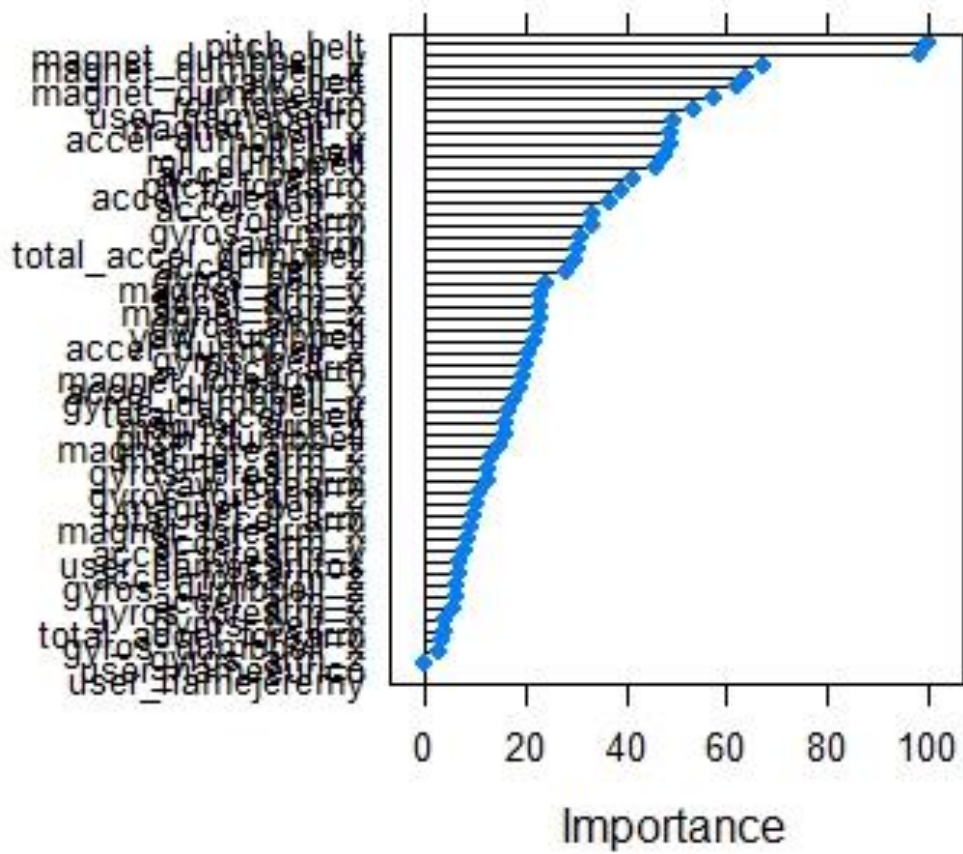
```
bwplot(results)
```

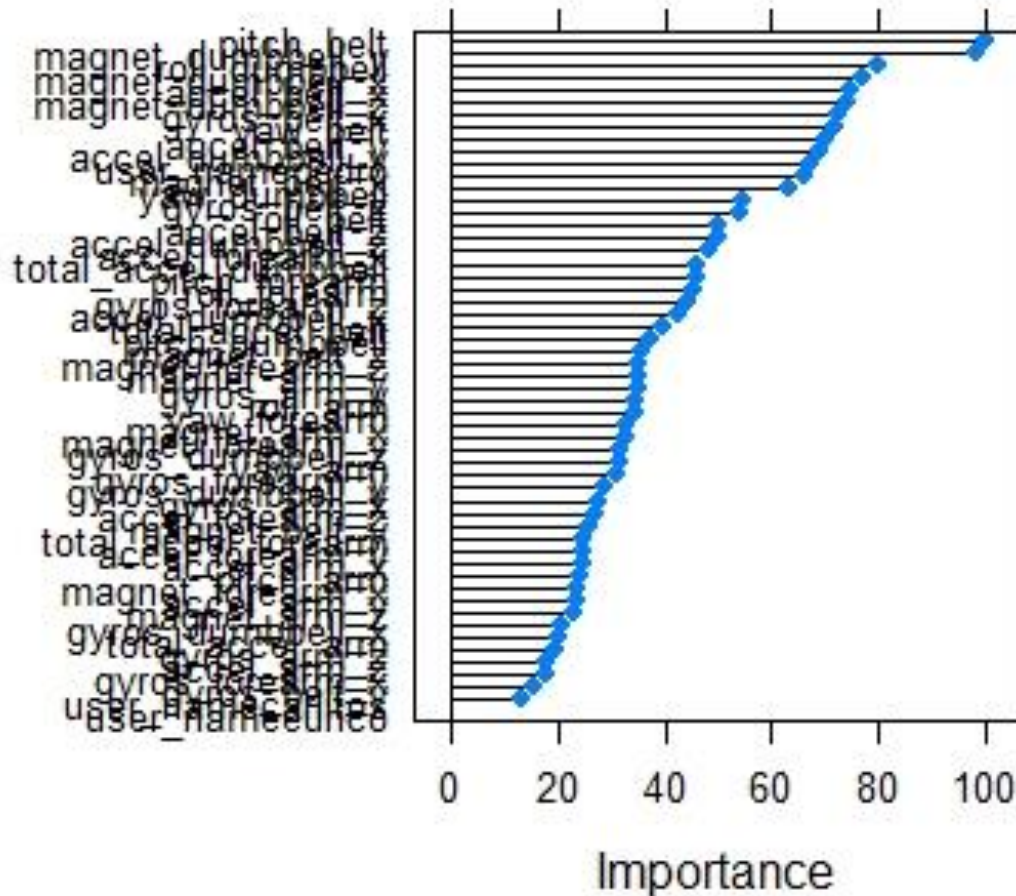
Dot-plot comparison

```
dotplot(results)
```

#e. Plot importance of variables

```
plot(varImp(rf_default))
plot(varImp(rf_gridsearch))
#
```





1. Expected Output

Solution report with commands, explanation of commands, and screenshots of the output should be submitted in .pdf format on GitHub the same GitHub should be expected to submit on student dashboard. This assignment contains 700 marks and will be evaluated within 14 days of submission.

Attempt 2

#-----Assignment 20-----

Perform the below given activities:

a. Create classification model using different random forest models

```
# b. Verify model goodness of fit
# c. Apply all the model validation techniques
# d. Make conclusions
# e. Plot importance of variables
```

```
#-----
```

```
# import data set
```

```
data_set <- weight_lifting
View(data_set)
```

```
# remove irrelevant collumns viz. name, cvtd_timestamp, new_window
data <- data_set[,-c(1,4,5)]
View(data)
str(data)
```

```
sum(is.na(data)) # there are no missing values
```

```
# splitting the data set for train and test
```

```
library(caTools)
set.seed(123)
split = sample.split(data$classe, SplitRatio = 0.7)
```

```
train = subset(data, split == TRUE)      # train data
test = subset(data, split == FALSE)      # test data
```

```
dim(train)
```

```
dim(test)
```

```
# a. Create classification model using different random forest models
```

```
library(tree); library(rpart); library(caret); library(C50); library(randomForest)
library(adabag); library(gbm)
```

```
# 1
train_control <- trainControl(method = "cv", number = 10)
cvmodel1 <- train(classe ~ ., data = train, trControl = train_control, method =
"rf")
cvpred1 <- predict(cvmodel1, test) # make prediction
cvconf1 <- confusionMatrix(test$classe, cvpred1) # confusion matrix
cvconf1$overall[1] # accuracy

# default
set.seed(123)
train_control <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
rf_default <- train(classe ~ ., data = train, trControl = train_control, method =
"rf",
metric = 'Accuracy', tuneGrid = expand.grid(.mtry =
sqrt(ncol(train))))
pred_rf_default <- predict(rf_default, test) # make prediction
conf_rf_default <- confusionMatrix(test$classe, pred_rf_default) # confusion
matrix
conf_rf_default$overall[1] # accuracy
varImp(rf_default) # var importance - 20

# random search for parameters
train_control <- trainControl(method = "repeatedcv", number = 10, repeats = 3,
search = 'random')
rf_random <- train(classe ~ ., data = train, trControl = train_control, method =
"rf",
metric = 'Accuracy', tuneLength = 15)
pred_rf_random <- predict(rf_random, test) # make prediction
conf_rf_random <- confusionMatrix(test$classe, pred_rf_random) #
confusion matrix
conf_rf_random$overall[1] # accuracy
varImp(rf_random) # var importance - 20

# Grid Search
```

```

train_control <- trainControl(method = "repeatedcv", number = 10, repeats = 3,
search = 'grid')
rf_grid <- train(classe ~ ., data = train, trControl = train_control, method = "rf",
metric = 'Accuracy', tuneGrid = expand.grid(.mtry=c(1:15)))
pred_rf_grid <- predict(rf_grid, test) # make prediction
conf_rf_grid <- confusionMatrix(test$classe, pred_rf_grid) # confusion
matrix
conf_rf_grid$overall[1] # accuracy
varImp(rf_grid) # var importance - 20

```

Goodness of Fit

```

chisq.test(table(test$classe), prop.table(table(cvpred1))) # pv = 0.2202
chisq.test(table(test$classe), prop.table(table(pred_rf_default))) # pv = 0.2202
chisq.test(table(test$classe), prop.table(table(pred_rf_random))) # pv = 0.2202
chisq.test(table(test$classe), prop.table(table(pred_rf_grid))) # pv = 0.2202

```

Problem was to predict how well the activity is performed

The target variable is the 5 classe; 1 accurate and 4 type of error

occurred during the activity

error (target) detection was done by classifying an

execution to one of the mistake classes

we could detect mistakes fairly accurately

Gradient boosting model is most accurate with less number of predictors

Model is good fit and the Accuracy is 1

```

plot <- plot(conf_rf_grid$table, col = topo.colors(6))

```

```

str(data)
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 4024 obs. of 156 variables:
 $ raw_timestamp_part_1 : num 1.32e+09 1.32e+09 1.32e+09 1.32e+09 1.32e+09 ...
 $ raw_timestamp_part_2 : num 34670 62641 70653 82654 90637 ...

```

```

$ num_window      : num  1 1 1 1 1 1 1 1 1 1 ...
$ roll_belt       : num  3.7 3.66 3.58 3.56 3.57 3.45 3.31 2.91 2.31 2 ...
$ pitch_belt      : num  41.6 42.8 43.7 44.4 45.1 45.6 46.2 46.9 47.4 47.7 ...
$ yaw_belt        : num  -82.8 -82.5 -82.3 -82.1 -81.9 -81.9 -81.9 -82.2 -82.6 -
$ total_accel_belt : num  3 2 1 1 1 1 3 4 2 3 ...
$ kurtosis_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_belt : chr  NA NA NA NA ...
$ kurtosis_yaw_belt : chr  NA NA NA NA ...
$ skewness_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ skewness_roll_belt_1 : chr  NA NA NA NA ...
$ skewness_yaw_belt : chr  NA NA NA NA ...
$ max_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_total_accel_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_belt    : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_belt : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_belt   : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_belt  : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_belt     : num  NA NA NA NA NA NA NA NA NA NA ...
$ gyros_belt_x     : num  2.02 1.96 1.88 1.8 1.77 1.75 1.78 1.75 1.65 1.48 ...
$ gyros_belt_y     : num  0.18 0.14 0.08 0.03 0 -0.03 -0.06 -0.06 -0.03 -0.06 ...
$ gyros_belt_z     : num  0.02 0.05 0.05 0.08 0.13 0.16 0.15 0.23 0.33 0.21 ...
$ accel_belt_x     : num  -3 -2 -2 -6 -4 1 1 2 -1 -18 ...
$ accel_belt_y     : num  -18 -13 -6 -5 -9 -9 -24 -36 -19 18 ...
$ accel_belt_z     : num  22 16 8 7 0 -5 -8 -9 -7 1 ...
$ magnet_belt_x    : num  387 405 409 422 418 432 438 440 443 449 ...
$ magnet_belt_y    : num  525 512 511 513 508 510 508 503 507 499 ...
$ magnet_belt_z    : num  -267 -254 -244 -221 -208 -189 -176 -163 -140 -132 ...
$ roll_arm         : num  132 129 125 120 115 110 104 98.6 93.2 88.5 ...
$ pitch_arm        : num  -43.7 -45.3 -46.8 -48.1 -49.1 -49.6 -49.9 -49.7 -49 -48
$ yaw_arm          : num  -53.6 -49 -43.7 -38.1 -31.7 -25.8 -18.5 -11.4 -4.49 1.8
$ total_accel_arm  : num  38 38 35 35 34 33 29 28 27 22 ...
$ var_accel_arm    : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_roll_arm     : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_roll_arm  : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_roll_arm     : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_pitch_arm    : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_pitch_arm : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_pitch_arm    : num  NA NA NA NA NA NA NA NA NA NA ...
$ avg_yaw_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
$ stddev_yaw_arm   : num  NA NA NA NA NA NA NA NA NA NA ...
$ var_yaw_arm      : num  NA NA NA NA NA NA NA NA NA NA ...
$ gyros_arm_x      : num  2.65 2.79 2.91 3.08 3.2 3.31 3.5 3.53 3.4 3.48 ...
$ gyros_arm_y      : num  -0.61 -0.64 -0.69 -0.72 -0.77 -0.83 -0.83 -0.83 -0.83 -
$ gyros_arm_z      : num  -0.02 -0.11 -0.15 -0.23 -0.25 -0.3 -0.31 -0.21 -0.11 -0

```

```

$ accel_arm_x      : num 143 146 156 158 163 160 165 153 143 135 ...
$ accel_arm_y      : num 30 35 44 52 55 59 67 70 78 96 ...
$ accel_arm_z      : num -346 -339 -307 -305 -288 -274 -225 -218 -205 -134 ...
$ magnet_arm_x      : num 556 599 613 646 670 696 721 725 740 741 ...
$ magnet_arm_y      : num -205 -206 -198 -186 -175 -174 -161 -152 -133 -115 ...
$ magnet_arm_z      : num -374 -335 -319 -268 -241 -193 -121 -105 -43 14 ...
$ kurtosis_roll_arm : chr NA NA NA NA ...
$ kurtosis_pitch_arm : chr NA NA NA NA ...
$ kurtosis_yaw_arm   : chr NA NA NA NA ...
$ skewness_roll_arm  : chr NA NA NA NA ...
$ skewness_pitch_arm : chr NA NA NA NA ...
$ skewness_yaw_arm   : chr NA NA NA NA ...
$ max_roll_arm       : num NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_arm      : num NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_arm        : num NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_arm       : num NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_arm      : num NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_arm        : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_arm : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_arm : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_arm  : num NA NA NA NA NA NA NA NA NA NA ...
$ roll_dumbbell      : num 51.2 55.8 55.5 55.9 55.2 ...
$ pitch_dumbbell     : num 11.7 9.65 6.88 11.08 11.43 ...
$ yaw_dumbbell       : num 104.3 100.2 101.1 99.8 100.4 ...
$ kurtosis_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ kurtosis_yaw_dumbbell : chr NA NA NA NA ...
$ skewness_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ skewness_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ skewness_yaw_dumbbell : chr NA NA NA NA ...
$ max_roll_dumbbell  : num NA NA NA NA NA NA NA NA NA NA ...
$ max_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ max_yaw_dumbbell   : num NA NA NA NA NA NA NA NA NA NA ...
$ min_roll_dumbbell  : num NA NA NA NA NA NA NA NA NA NA ...
$ min_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ min_yaw_dumbbell   : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_roll_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_pitch_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ amplitude_yaw_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
$ total_accel_dumbbell : num 4 4 4 5 4 4 4 4 4 4 ...
$ var_accel_dumbbell : num NA NA NA NA NA NA NA NA NA NA ...
[list output truncated]
- attr(*, "spec")=List of 3
..$ cols :List of 159
.. ..$ user_name : list()
.. .. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ raw_timestamp_part_1 : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ raw_timestamp_part_2 : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ cvtd_timestamp : list()
.. .. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ new_window : list()
.. .. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ num_window : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"

```

```
.. ..$ roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ total_accel_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_belt_1 : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_total_accel_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_roll_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ var_pitch_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ avg_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ stddev_yaw_belt : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
```

```
.. ..$ var_yaw_belt      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ gyros_belt_x      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ gyros_belt_y      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ gyros_belt_z      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ accel_belt_x      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ accel_belt_y      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ accel_belt_z      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ magnet_belt_x     : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ magnet_belt_y     : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ magnet_belt_z     : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ roll_arm          : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ pitch_arm         : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ yaw_arm           : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ total_accel_arm   : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ var_accel_arm     : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ avg_roll_arm      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ stddev_roll_arm   : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ var_roll_arm      : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ avg_pitch_arm     : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ stddev_pitch_arm  : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ var_pitch_arm     : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ avg_yaw_arm       : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ stddev_yaw_arm    : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ var_yaw_arm       : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ gyros_arm_x       : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ gyros_arm_y       : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ gyros_arm_z       : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
.. ..$ accel_arm_x       : list()
.. ..  .- attr(*, "class")= chr  "collector_double" "collector"
```



```
.. ..$ accel_arm_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ accel_arm_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_x      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_y      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ magnet_arm_z      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_arm  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_yaw_arm  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_roll_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_arm      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_arm     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_arm       : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_arm : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_yaw_arm  : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ roll_dumbbell     : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ pitch_dumbbell    : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ yaw_dumbbell      : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ kurtosis_yaw_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_roll_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ skewness_pitch_dumbbell : list()
.. ..- attr(*, "class")= chr "collector_double" "collector"
```

```

.. ..$ skewness_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_character" "collector"
.. ..$ max_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ max_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ min_yaw_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_roll_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. ..$ amplitude_pitch_dumbbell : list()
.. .. ..- attr(*, "class")= chr "collector_double" "collector"
.. .. [list output truncated]
..$ default: list()
.. ..- attr(*, "class")= chr "collector_guess" "collector"
..$ skip : num 1
.. ..- attr(*, "class")= chr "col_spec"
> sum(is.na(data)) # there are no missing values
[1] 393600
>

```

```
> set.seed(123)
> split = sample.split(data$classe, SplitRatio = 0.7)
> train = subset(data, split == TRUE)           # train data
> test = subset(data, split == FALSE)           # test data
> dim(train)
[1] 2816 156
```

```
> dim(test)
[1] 1208 156
```

```
> # 1
> train_control <- trainControl(method = "cv", number = 10)
> cvmodel1 <- train(classe ~ ., data = train, trControl = train_control, method = "rf")
Error in na.fail.default(list(classe = c("E", "E", "E", "E", "E", "E", :
  missing values in object
> cvpred1 <- predict(cvmodel1, test) # make prediction
```

```

Error in predict(cvmodell, test) : object 'cvmodell' not found
> cvconf1 <- confusionMatrix(test$classe, cvpred1)           # confusion matrix
Error in is.factor(reference) : object 'cvpred1' not found
> cvconf1$overall[1]                                         # accuracy
Error: object 'cvconf1' not found
> # default
> set.seed(123)
> train_control <- trainControl(method = "repeatedcv", number = 10, repeats
= 3)
> rf_default <- train(classe ~ ., data = train, trControl = train_control, method = "rf",
+                       metric = 'Accuracy', tuneGrid = expand.grid(.mtry = sqrt(ncol(train))))
Error in na.fail.default(list(classe = c("E", "E", "E", "E", "E", "E", :
missing values in object
> pred_rf_default <- predict(rf_default, test)               #
make prediction
Error in eval(predvars, data, env) : object 'user_name' not found
> conf_rf_default <- confusionMatrix(test$classe, pred_rf_default) #
confusion matrix
Error in is.factor(reference) : object 'pred_rf_default' not found
> conf_rf_default$overall[1]                                 #
accuracy
Error: object 'conf_rf_default' not found
> varImp(rf_default)                                         #
var importance - 20
rf variable importance

only 20 most important variables shown (out of 56)

pitch_belt          Overall
magnet_dumbbell_y    98.23
magnet_dumbbell_z    67.39
yaw_belt             63.92
magnet_dumbbell_x    62.26
roll_forearm         57.59
user_namepedro       53.43
magnet_belt_x         49.19
accel_dumbbell_y      48.96
roll_belt            48.78
roll_dumbbell        47.85
accel_belt_x         45.83
pitch_forearm        41.60
accel_forearm_x       39.22
accel_belt_y         36.96
roll_arm             33.60
gyros_arm_y          33.58
yaw_arm              30.79
total_accel_dumbbell 30.46
accel_belt_z         29.99

```

End >>>