**Cab Fare Prediction Solution Using R**

**Report by AMAN ARYA**

**Index**

1. Load libraries
2. Load packages
3. Loading datasets
4. Structure of data
5. Exploratory Data Analysis
6. Mode method
7. Mean Method
8. Median method
9. Knn Imputation
10. Outlier Analysis
11. Feature Engineering
12. Feature selection
13. Feature Scaling
14. Normalisation
15. Linear regression
16. Decision Tree
17. Random forest
18. Submission

**Loading Libraries**

rm(list= ls())

setwd(“C:/Users/DELL/Desktop/Cab Fare Prediction”)

x = c("ggplot2", "corrgram", "DMwR", "usdm", "caret", "randomForest", "e1071",

"DataCombine", "doSNOW", "inTrees", "rpart.plot", "rpart",'MASS','xgboost','stats')

**Load packages**

lapply(x, require, character.only = TRUE)

rm(x)

# The details of data attributes in the dataset are as follows:

# pickup\_datetime - timestamp value indicating when the cab ride started.

# pickup\_longitude - float for longitude coordinate of where the cab ride started.

# pickup\_latitude - float for latitude coordinate of where the cab ride started.

# dropoff\_longitude - float for longitude coordinate of where the cab ride ended.

# dropoff\_latitude - float for latitude coordinate of where the cab ride ended.

# passenger\_count - an integer indicating the number of passengers in the cab ride.

**Loading datasets**

train = read.csv("C:/Users/DELL/Desktop/data/train\_cab.csv", header = T, na.strings = c(" ", "", "NA"))

test = read.csv("C:/Users/DELL/Desktop/data/test.csv")

test\_pickup\_datetime = test["pickup\_datetime"]

**Structure of data**

str(train)

'data.frame': 16067 obs. of 7 variables:

$ fare\_amount : Factor w/ 468 levels "-2.5","-2.9",..: 301 58 373 432 370 26 431 56 NA 453 ...

$ pickup\_datetime : Factor w/ 16021 levels "2009-01-01 01:31:49 UTC",..: 1115 2509 6550 8252 2919 4986 9743 7479 9838 1606 ...

$ pickup\_longitude : num -73.8 -74 -74 -74 -74 ...

$ pickup\_latitude : num 40.7 40.7 40.8 40.7 40.8 ...

$ dropoff\_longitude: num -73.8 -74 -74 -74 -74 ...

$ dropoff\_latitude : num 40.7 40.8 40.8 40.8 40.8 ...

$ passenger\_count : num 1 1 2 1 1 1 1 1 1 2 ...

str(test)

'data.frame': 9914 obs. of 6 variables:

$ pickup\_datetime : Factor w/ 1753 levels "2009-01-01 11:04:24 UTC",..: 1648 1648 747 1041 1041 1041 744 744 744 1384 ...

$ pickup\_longitude : num -74 -74 -74 -74 -74 ...

$ pickup\_latitude : num 40.8 40.7 40.8 40.8 40.8 ...

$ dropoff\_longitude: num -74 -74 -74 -74 -74 ...

$ dropoff\_latitude : num 40.7 40.7 40.7 40.8 40.7 ...

$ passenger\_count : int 1 1 1 1 1 1 1 1 1 1 ...

summary(train)

fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude

6.5 : 759 2009-04-18 20:44:00 UTC: 2 Min. :-74.44 Min. :-74.01

4.5 : 671 2009-05-10 17:57:00 UTC: 2 1st Qu.:-73.99 1st Qu.: 40.73

8.5 : 630 2009-07-01 15:55:00 UTC: 2 Median :-73.98 Median : 40.75

5.3 : 475 2009-07-28 13:37:00 UTC: 2 Mean :-72.46 Mean : 39.91

4.9 : 464 2009-12-10 15:37:00 UTC: 2 3rd Qu.:-73.97 3rd Qu.: 40.77

(Other):13044 2009-12-11 11:56:00 UTC: 2 Max. : 40.77 Max. :401.08

NA's : 24 (Other) :16055

dropoff\_longitude dropoff\_latitude passenger\_count

Min. :-74.43 Min. :-74.01 Min. : 0.000

1st Qu.:-73.99 1st Qu.: 40.73 1st Qu.: 1.000

Median :-73.98 Median : 40.75 Median : 1.000

Mean :-72.46 Mean : 39.90 Mean : 2.625

3rd Qu.:-73.96 3rd Qu.: 40.77 3rd Qu.: 2.000

Max. : 40.80 Max. : 41.37 Max. :5345.000

NA's :55

summary(test)

pickup\_datetime pickup\_longitude pickup\_latitude dropoff\_longitude

2011-12-13 22:00:00 UTC: 270 Min. :-74.25 Min. :40.57 Min. :-74.26

2013-09-25 22:00:00 UTC: 251 1st Qu.:-73.99 1st Qu.:40.74 1st Qu.:-73.99

2012-11-20 21:54:00 UTC: 246 Median :-73.98 Median :40.75 Median :-73.98

2014-07-21 18:19:00 UTC: 243 Mean :-73.97 Mean :40.75 Mean :-73.97

2010-08-27 18:45:00 UTC: 235 3rd Qu.:-73.97 3rd Qu.:40.77 3rd Qu.:-73.96

2011-06-01 07:37:00 UTC: 227 Max. :-72.99 Max. :41.71 Max. :-72.99

(Other) :8442

dropoff\_latitude passenger\_count

Min. :40.57 Min. :1.000

1st Qu.:40.74 1st Qu.:1.000

Median :40.75 Median :1.000

Mean :40.75 Mean :1.671

3rd Qu.:40.77 3rd Qu.:2.000

Max. :41.70 Max. :6.000

head(train,5)

fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude

1 4.5 2009-06-15 17:26:21 UTC -73.84431 40.72132

2 16.9 2010-01-05 16:52:16 UTC -74.01605 40.71130

3 5.7 2011-08-18 00:35:00 UTC -73.98274 40.76127

4 7.7 2012-04-21 04:30:42 UTC -73.98713 40.73314

5 5.3 2010-03-09 07:51:00 UTC -73.96810 40.76801

dropoff\_longitude dropoff\_latitude passenger\_count

1 -73.84161 40.71228 1

2 -73.97927 40.78200 1

3 -73.99124 40.75056 2

4 -73.99157 40.75809 1

5 -73.95665 40.78376 1

head(test,5)

pickup\_datetime pickup\_longitude pickup\_latitude dropoff\_longitude

1 2015-01-27 13:08:24 UTC -73.97332 40.76381 -73.98143

2 2015-01-27 13:08:24 UTC -73.98686 40.71938 -73.99889

3 2011-10-08 11:53:44 UTC -73.98252 40.75126 -73.97965

4 2012-12-01 21:12:12 UTC -73.98116 40.76781 -73.99045

5 2012-12-01 21:12:12 UTC -73.96605 40.78977 -73.98856

dropoff\_latitude passenger\_count

1 40.74384 1

2 40.73920 1

3 40.74614 1

4 40.75164 1

5 40.74443 1

**Exploratory Data Analysis**

train$fare\_amount = as.numeric(as.character(train$fare\_amount))

train$passenger\_count=round(train$passenger\_count)

train[which(train$fare\_amount < 1 ),]

fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude

2040 -2.90 2010-03-09 23:37:10 UTC -73.78945 40.64350

2487 -2.50 2015-03-22 05:14:27 UTC -74.00003 40.72063

2781 0.01 2015-05-01 15:38:41 UTC -73.93904 40.71396

10003 0.00 2010-02-15 14:26:01 UTC -73.98712 40.73881

13033 -3.00 2013-08-30 08:57:10 UTC -73.99506 40.74076

dropoff\_longitude dropoff\_latitude passenger\_count

2040 -73.78866 40.64195 1

2487 -73.99981 40.72054 1

2781 -73.94167 40.71400 1

10003 -74.00591 40.71396 1

13033 -73.99589 40.74136 4

nrow(train[which(train$fare\_amount < 1 ),])

[1] 5

train = train[-which(train$fare\_amount < 1 ),]

for (i in seq(4,11,by=1)){

print(paste('passenger\_count above ' ,i,nrow(train[which(train$passenger\_count > i ),])))

}

[1] "passenger\_count above 4 1367"

[1] "passenger\_count above 5 322"

[1] "passenger\_count above 6 20"

[1] "passenger\_count above 7 20"

[1] "passenger\_count above 8 20"

[1] "passenger\_count above 9 20"

[1] "passenger\_count above 10 20"

[1] "passenger\_count above 11 20"

train[which(train$passenger\_count > 6 ),]

fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude

234 8.5 2011-07-24 01:14:35 UTC 0.00000 0.00000

264 4.9 2010-07-12 09:44:33 UTC -73.98325 40.73465

294 6.1 2011-01-18 23:48:00 UTC -74.00664 40.73893

357 8.5 2013-06-18 10:27:05 UTC -73.99211 40.76420

387 8.1 2009-08-21 19:35:05 UTC -73.96085 40.76156

414 NA 2013-09-12 11:32:00 UTC -73.98206 40.77271

972 10.1 2010-11-21 01:41:00 UTC -74.00450 40.74214

1008 3.7 2010-12-14 14:46:00 UTC -73.96916 40.75900

1044 5.7 2012-08-22 22:08:29 UTC -73.97357 40.76018

1108 4.9 2009-08-08 21:50:50 UTC -73.98898 40.72107

1147 8.0 2014-03-27 08:05:01 UTC -73.99110 40.77065

1201 9.7 2011-08-16 09:29:00 UTC -73.98049 40.74161

1243 5.3 2011-10-16 00:22:00 UTC -73.98109 40.73816

8407 6.9 2010-08-25 11:41:00 UTC 0.00000 0.00000

8446 5.7 2009-03-28 22:00:00 UTC -73.98241 40.75132

8507 11.3 2010-05-23 20:06:37 UTC -73.98542 40.73847

8572 12.5 2011-12-03 03:21:00 UTC -73.99372 40.76204

8632 20.0 2012-12-10 22:28:00 UTC -73.95544 40.67023

8716 4.5 2009-09-04 09:14:03 UTC -73.97752 40.75848

8986 8.5 2015-01-14 15:10:21 UTC -73.95544 40.78761

dropoff\_longitude dropoff\_latitude passenger\_count

234 0.00000 0.00000 236

264 -73.99128 40.73892 456

294 -74.01083 40.71791 5334

357 -73.97300 40.76270 535

387 -73.97634 40.74836 354

414 -73.95621 40.77178 55

972 -73.99433 40.72041 554

1008 -73.96876 40.76462 53

1044 -73.95356 40.76739 35

1108 -73.98237 40.73206 345

1147 -73.97693 40.79007 5345

1201 -73.98062 40.74687 536

1243 -73.99059 40.74010 43

8407 0.00000 0.00000 53

8446 -73.97129 40.74850 58

8507 -74.00170 40.70776 537

8572 -73.97753 40.73402 87

8632 -74.00480 40.73148 43

8716 -73.98325 40.74984 531

8986 -73.96556 40.79869 557

train[which(train$passenger\_count <1 ),]

fare\_amount pickup\_datetime pickup\_longitude pickup\_latitude

315 34.0 2015-06-02 23:16:15 UTC -73.97490 40.75109

567 4.9 2012-01-28 21:33:18 UTC -73.95532 40.78284

679 6.5 2012-02-27 07:24:20 UTC -73.98340 40.73818

1161 13.3 2011-05-25 23:58:48 UTC -73.99836 40.74035

1936 10.1 2011-10-23 11:09:28 UTC -73.97140 40.79500

2201 8.1 2011-05-23 16:54:19 UTC -73.98801 40.74830

2426 8.9 2011-11-25 22:47:33 UTC -73.99990 40.73860

3035 5.7 2011-03-06 12:03:14 UTC -73.98656 40.74578

3414 7.3 2011-02-28 06:39:16 UTC -73.97341 40.74371

3482 11.3 2011-11-30 17:23:02 UTC -73.96810 40.76250

3490 7.3 2011-10-18 08:18:18 UTC -73.96610 40.79450

3600 8.9 2011-03-06 18:24:50 UTC -73.98266 40.74614

4115 4.5 2011-07-22 05:51:00 UTC -73.97550 40.76080

4249 4.1 2012-02-15 11:19:50 UTC -73.98211 40.72107

4345 24.9 2011-05-19 13:28:17 UTC -73.78923 40.64662

4355 5.3 2012-01-15 12:06:43 UTC -73.96440 40.76750

5059 12.9 2011-11-05 18:29:30 UTC -74.00890 40.70940

5151 7.7 2011-11-07 22:07:24 UTC -73.98490 40.67530

5162 11.3 2011-12-20 06:59:57 UTC -73.98370 40.77580

5278 6.1 2012-04-12 09:35:22 UTC -73.96700 40.77242

5518 3.3 2012-07-02 16:11:55 UTC -74.00711 40.74386

5558 27.3 2011-02-08 13:31:18 UTC -73.87332 40.77395

5689 4.9 2011-09-07 21:50:40 UTC -73.98670 40.76130

5915 8.5 2011-12-19 22:44:42 UTC -73.98200 40.75590

6037 7.7 2011-02-26 12:41:03 UTC -73.97671 40.77571

6576 4.5 2012-04-19 19:44:48 UTC -73.96460 40.80721

6714 10.5 2012-03-06 15:35:37 UTC -73.96511 40.76646

6882 5.7 2011-06-14 10:12:16 UTC -73.95900 40.78100

7280 10.9 2012-04-18 18:44:15 UTC -73.98293 40.72264

7521 6.9 2011-06-08 13:11:10 UTC -73.97510 40.75520

7641 10.9 2012-01-21 16:17:50 UTC -73.99950 40.72500

7910 6.9 2011-03-12 17:03:23 UTC -73.97479 40.75986

8322 7.3 2012-02-25 19:19:41 UTC -74.00373 40.74211

8662 8.1 2012-01-13 13:47:57 UTC -73.99970 40.72180

8863 11.5 2013-05-17 07:15:00 UTC -73.96984 40.75302

8917 2.5 2011-09-13 18:45:31 UTC -73.78330 40.64860

8972 7.7 2011-10-22 20:50:34 UTC -73.99860 40.76100

9160 3.3 2011-06-04 10:38:47 UTC -73.98730 40.72920

9966 9.3 2012-01-04 22:04:14 UTC -73.97100 40.75490

10643 10.1 2012-03-10 18:21:16 UTC -73.97227 40.75404

10664 18.1 2011-05-06 13:38:48 UTC -73.97382 40.78936

10712 6.5 2012-03-31 20:16:01 UTC -74.00432 40.72407

11463 15.7 2011-10-20 23:09:45 UTC -73.99480 40.75040

11804 12.1 2011-06-21 08:58:36 UTC -73.99900 40.72480

12217 5.7 2012-03-27 16:06:09 UTC -73.98202 40.75634

12612 8.1 2011-07-22 23:13:58 UTC -73.97800 40.75230

13030 4.5 2012-02-01 21:18:24 UTC -73.98212 40.77041

13228 10.9 2011-04-22 18:13:12 UTC -73.99951 40.72208

13380 7.3 2012-03-14 07:20:10 UTC -73.96451 40.77158

13715 4.1 2011-12-15 07:05:33 UTC -73.97940 40.73110

13743 21.7 2011-04-29 12:51:14 UTC -74.00947 40.70214

14197 8.1 2011-10-06 23:31:04 UTC -73.99940 40.74380

14309 8.1 2011-11-11 12:03:38 UTC 0.00000 0.00000

14873 3.3 2011-03-02 19:25:46 UTC -73.94855 40.77397

15287 4.9 2012-03-20 22:23:34 UTC -73.97181 40.76014

15515 6.5 2012-02-12 02:03:50 UTC -73.99934 40.71887

15555 6.5 2011-08-03 08:31:19 UTC -73.99170 40.75010

15920 16.5 2011-10-17 08:58:54 UTC 0.00000 0.00000

dropoff\_longitude dropoff\_latitude passenger\_count

315 -73.90855 40.88188 0

567 -73.95580 40.77367 0

679 -73.97140 40.75802 0

1161 -73.94646 40.77735 0

1936 -73.96790 40.76860 0

2201 -74.00518 40.73873 0

2426 -73.97180 40.74630 0

3035 -73.99455 40.73000 0

3414 -73.98522 40.74158 0

3482 -73.98440 40.76090 0

3490 -73.96030 40.77980 0

3600 -74.00558 40.72440 0

4115 -73.99020 40.76000 0

4249 -73.99224 40.72531 0

4345 -73.72503 40.61414 0

4355 -73.98160 40.77400 0

5059 -73.98540 40.75620 0

5151 -74.01040 40.65520 0

5162 -73.98530 40.74130 0

5278 -73.96886 40.76115 0

5518 -74.00334 40.74888 0

5558 -74.01020 40.71116 0

5689 -73.98230 40.77390 0

5915 -73.95560 40.76960 0

6037 -73.97279 40.76447 0

6576 -73.97088 40.79796 0

6714 -73.99752 40.74488 0

6882 -73.95030 40.77540 0

7280 -73.97111 40.76017 0

7521 -73.98000 40.76590 0

7641 -73.97326 40.76329 0

7910 -73.95572 40.77273 0

8322 -73.98827 40.75931 0

8662 -74.00560 40.74110 0

8863 -73.99858 40.71293 0

8917 -73.78330 40.64860 0

8972 -73.97790 40.77720 0

9160 -73.99400 40.73210 0

9966 -74.00450 40.73610 0

10643 -73.99409 40.73444 0

10664 -73.94225 40.75412 0

10712 -74.00645 40.73842 0

11463 -73.95920 40.71060 0

11804 -73.98060 40.75490 0

12217 -73.98099 40.74544 0

12612 -73.99180 40.76390 0

13030 -73.98006 40.76215 0

13228 -73.98950 40.74742 0

13380 -73.94559 40.77803 0

13715 -73.98150 40.72480 0

13743 -73.95919 40.78325 0

14197 -73.99040 40.72480 0

14309 -73.99000 40.75540 0

14873 -73.94510 40.77841 0

15287 -73.96203 40.76768 0

15515 -73.98411 40.72524 0

15555 -73.98100 40.75090 0

15920 -73.99970 40.73450 0

nrow(train[which(train$passenger\_count <1 ),])

[1] 58

train = train[-which(train$passenger\_count < 1 ),]

train = train[-which(train$passenger\_count > 6),]

print(paste('pickup\_longitude above 180=',nrow(train[which(train$pickup\_longitude >180 ),])))

[1] "pickup\_longitude above 180= 0"

print(paste('pickup\_longitude above -180=',nrow(train[which(train$pickup\_longitude < -180 ),])))

[1] "pickup\_longitude above -180= 0"

print(paste('pickup\_latitude above 90=',nrow(train[which(train$pickup\_latitude > 90 ),])))

[1] "pickup\_latitude above 90= 1"

print(paste('pickup\_latitude above -90=',nrow(train[which(train$pickup\_latitude < -90 ),])))

[1] "pickup\_latitude above -90= 0"

print(paste('dropoff\_longitude above 180=',nrow(train[which(train$dropoff\_longitude > 180 ),])))

[1] "dropoff\_longitude above 180= 0"

print(paste('dropoff\_longitude above -180=',nrow(train[which(train$dropoff\_longitude < -180 ),])))

[1] "dropoff\_longitude above -180= 0"

print(paste('dropoff\_latitude above -90=',nrow(train[which(train$dropoff\_latitude < -90 ),])))

[1] "dropoff\_latitude above -90= 0"

print(paste('dropoff\_latitude above 90=',nrow(train[which(train$dropoff\_latitude > 90 ),])))

[1] "dropoff\_latitude above 90= 0"

nrow(train[which(train$pickup\_longitude == 0 ),])

[1] 311

nrow(train[which(train$pickup\_latitude == 0 ),])

[1] 311

nrow(train[which(train$dropoff\_longitude == 0 ),])

[1] 312

nrow(train[which(train$pickup\_latitude == 0 ),])

[1] 311

train = train[-which(train$pickup\_latitude > 90),]

train = train[-which(train$pickup\_longitude == 0),]

train = train[-which(train$dropoff\_longitude == 0),]

df=train

**Missing Value Analysis**

missing\_val = data.frame(apply(train,2,function(x){sum(is.na(x))}))

missing\_val$Columns = row.names(missing\_val)

names(missing\_val)[1] = "Missing\_percentage"

missing\_val$Missing\_percentage = (missing\_val$Missing\_percentage/nrow(train)) \* 100

missing\_val = missing\_val[order(-missing\_val$Missing\_percentage),]

row.names(missing\_val) = NULL

missing\_val = missing\_val[,c(2,1)]

missing\_val

Columns Missing\_percentage

1 passenger\_count 0.3511909

2 fare\_amount 0.1404763

3 pickup\_datetime 0.0000000

4 pickup\_longitude 0.0000000

5 pickup\_latitude 0.0000000

6 dropoff\_longitude 0.0000000

7 dropoff\_latitude 0.0000000

unique(train$passenger\_count)

[1] 1 2 3 NA 6 5 4

unique(test$passenger\_count)

[1] 1 2 3 4 5 6

train[,'passenger\_count'] = factor(train[,'passenger\_count'], labels=(1:6))

test[,'passenger\_count'] = factor(test[,'passenger\_count'], labels=(1:6))

train$passenger\_count[1000]

[1] 1

Levels: 1 2 3 4 5 6

train$passenger\_count[1000] = NA

getmode <- function(v) {

uniqv <- unique(v)

uniqv[which.max(tabulate(match(v, uniqv)))]

}

**Mode Method**

getmode(train$passenger\_count)

[1] 1

Levels: 1 2 3 4 5 6

sapply(train, sd, na.rm = TRUE)

train$fare\_amount[1000]

[1] 18.1

train$fare\_amount[1000]= NA

**Mean Method**

mean(train$fare\_amount, na.rm = T)

[1] 15.11749

**Median Method**

median(train$fare\_amount, na.rm = T)

[1] 8.5

**kNN Imputation**

train = knnImputation(train, k = 181)

train$fare\_amount[1000]

[1] 18.57022

train$passenger\_count[1000]

[1] 1

Levels: 1 2 3 4 5 6

sapply(train, sd, na.rm = TRUE)

sum(is.na(train))

[1] 0

str(train)

'data.frame': 15661 obs. of 7 variables:

$ fare\_amount : num 4.5 16.9 5.7 7.7 5.3 ...

$ pickup\_datetime : Factor w/ 16021 levels "2009-01-01 01:31:49 UTC",..: 1115 2509 6550 8252 2919 4986 9743 7479 9838 1606 ...

$ pickup\_longitude : num -73.8 -74 -74 -74 -74 ...

$ pickup\_latitude : num 40.7 40.7 40.8 40.7 40.8 ...

$ dropoff\_longitude: num -73.8 -74 -74 -74 -74 ...

$ dropoff\_latitude : num 40.7 40.8 40.8 40.8 40.8 ...

$ passenger\_count : Factor w/ 6 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 2 ...

summary(train)

fare\_amount pickup\_datetime pickup\_longitude

Min. : 1.14 2009-04-18 20:44:00 UTC: 2 Min. :-74.44

1st Qu.: 6.00 2009-05-10 17:57:00 UTC: 2 1st Qu.:-73.99

Median : 8.50 2009-07-01 15:55:00 UTC: 2 Median :-73.98

Mean : 15.11 2009-07-28 13:37:00 UTC: 2 Mean :-73.91

3rd Qu.: 12.50 2009-12-10 15:37:00 UTC: 2 3rd Qu.:-73.97

Max. :54343.00 2009-12-11 11:56:00 UTC: 2 Max. : 40.77

(Other) :15649

pickup\_latitude dropoff\_longitude dropoff\_latitude passenger\_count

Min. :-74.01 Min. :-74.43 Min. :-74.01 1:11071

1st Qu.: 40.74 1st Qu.:-73.99 1st Qu.: 40.74 2: 2286

Median : 40.75 Median :-73.98 Median : 40.75 3: 663

Mean : 40.69 Mean :-73.91 Mean : 40.69 4: 320

3rd Qu.: 40.77 3rd Qu.:-73.97 3rd Qu.: 40.77 5: 1025

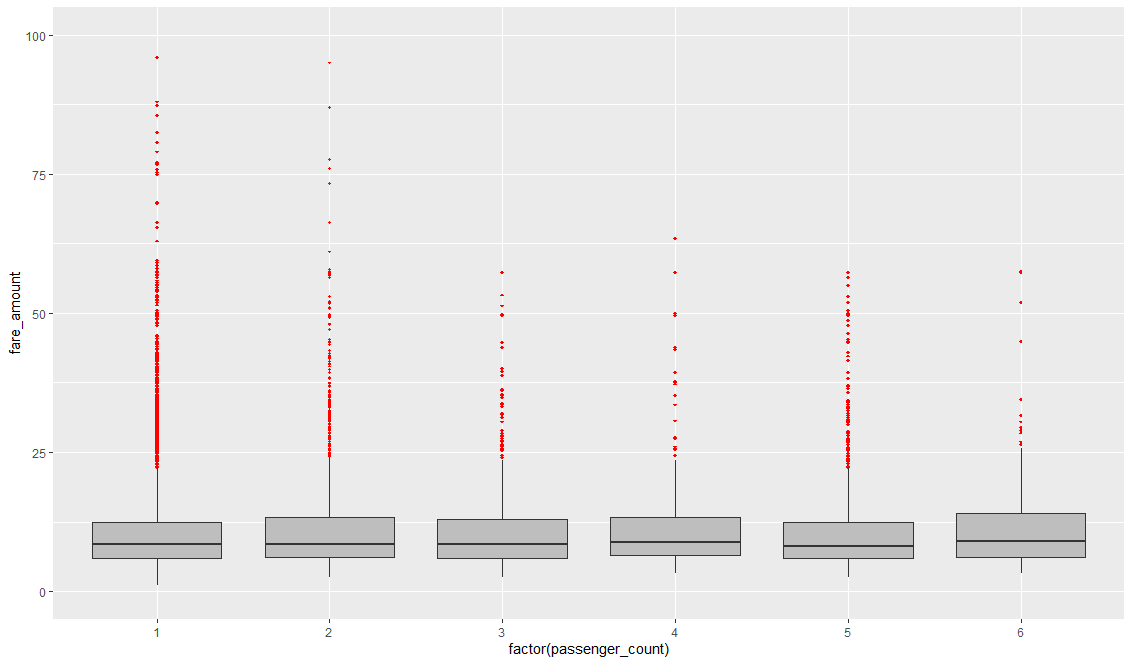
Max. : 41.37 Max. : 40.80 Max. : 41.37 6: 296

df1=train

**Outlier Analysis**

pl1 = ggplot(train,aes(x = factor(passenger\_count),y = fare\_amount))

pl1 + geom\_boxplot(outlier.colour="red", fill = "grey" ,outlier.shape=18,outlier.size=1, notch=FALSE)+ylim(0,100)



vals = train[,"fare\_amount"] %in% boxplot.stats(train[,"fare\_amount"])$out

train[which(vals),"fare\_amount"] = NA

sum(is.na(train$fare\_amount))

[1] 1358

train = knnImputation(train,k=3)

sum(is.na(train$fare\_amount))

[1] 0

str(train)

'data.frame': 15661 obs. of 7 variables:

$ fare\_amount : num 4.5 16.9 5.7 7.7 5.3 ...

$ pickup\_datetime : Factor w/ 16021 levels "2009-01-01 01:31:49 UTC",..: 1115 2509 6550 8252 2919 4986 9743 7479 9838 1606 ...

$ pickup\_longitude : num -73.8 -74 -74 -74 -74 ...

$ pickup\_latitude : num 40.7 40.7 40.8 40.7 40.8 ...

$ dropoff\_longitude: num -73.8 -74 -74 -74 -74 ...

$ dropoff\_latitude : num 40.7 40.8 40.8 40.8 40.8 ...

$ passenger\_count : Factor w/ 6 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 2 ...

df2=train

**Feature Engineering**

train$pickup\_date = as.Date(as.character(train$pickup\_datetime))

train$pickup\_weekday = as.factor(format(train$pickup\_date,"%u"))# Monday = 1

train$pickup\_mnth = as.factor(format(train$pickup\_date,"%m"))

train$pickup\_yr = as.factor(format(train$pickup\_date,"%Y"))

pickup\_time = strptime(train$pickup\_datetime,"%Y-%m-%d %H:%M:%S")

train$pickup\_hour = as.factor(format(pickup\_time,"%H"))

test$pickup\_date = as.Date(as.character(test$pickup\_datetime))

test$pickup\_weekday = as.factor(format(test$pickup\_date,"%u"))# Monday = 1

test$pickup\_mnth = as.factor(format(test$pickup\_date,"%m"))

test$pickup\_yr = as.factor(format(test$pickup\_date,"%Y"))

pickup\_time = strptime(test$pickup\_datetime,"%Y-%m-%d %H:%M:%S")

test$pickup\_hour = as.factor(format(pickup\_time,"%H"))

sum(is.na(train))

[1] 5

train = na.omit(train)

train = subset(train,select = -c(pickup\_datetime,pickup\_date))

test = subset(test,select = -c(pickup\_datetime,pickup\_date))

deg\_to\_rad = function(deg){

(deg \* pi) / 180

}

haversine = function(long1,lat1,long2,lat2){

#long1rad = deg\_to\_rad(long1)

phi1 = deg\_to\_rad(lat1)

#long2rad = deg\_to\_rad(long2)

phi2 = deg\_to\_rad(lat2)

delphi = deg\_to\_rad(lat2 - lat1)

dellamda = deg\_to\_rad(long2 - long1)

a = sin(delphi/2) \* sin(delphi/2) + cos(phi1) \* cos(phi2) \*

sin(dellamda/2) \* sin(dellamda/2)

c = 2 \* atan2(sqrt(a),sqrt(1-a))

R = 6371e3

R \* c / 1000 #1000 is used to convert to meters

}

train$dist = haversine(train$pickup\_longitude,train$pickup\_latitude,train$dropoff\_longitude,train$dropoff\_latitude)

test$dist = haversine(test$pickup\_longitude,test$pickup\_latitude,test$dropoff\_longitude,test$dropoff\_latitude)

train = subset(train,select = -c(pickup\_longitude,pickup\_latitude,dropoff\_longitude,dropoff\_latitude))

test = subset(test,select = -c(pickup\_longitude,pickup\_latitude,dropoff\_longitude,dropoff\_latitude))

str(train)

'data.frame': 15660 obs. of 7 variables:

$ fare\_amount : num 4.5 16.9 5.7 7.7 5.3 ...

$ passenger\_count: Factor w/ 6 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 1 2 ...

$ pickup\_weekday : Factor w/ 7 levels "1","2","3","4",..: 1 2 4 6 2 4 2 3 1 3 ...

$ pickup\_mnth : Factor w/ 12 levels "01","02","03",..: 6 1 8 4 3 1 11 1 12 9 ...

$ pickup\_yr : Factor w/ 7 levels "2009","2010",..: 1 2 3 4 2 3 4 4 4 1 ...

$ pickup\_hour : Factor w/ 24 levels "00","01","02",..: 18 17 1 5 8 10 21 18 14 2 ...

$ dist : num 1.03 8.45 1.39 2.8 2 ...

summary(train)

fare\_amount passenger\_count pickup\_weekday pickup\_mnth pickup\_yr

Min. : 1.14 1:11070 1:2079 05 :1476 2009:2429

1st Qu.: 6.00 2: 2286 2:2224 03 :1473 2010:2432

Median : 8.50 3: 663 3:2256 06 :1461 2011:2413

Mean : 9.40 4: 320 4:2269 01 :1435 2012:2500

3rd Qu.:12.00 5: 1025 5:2323 04 :1406 2013:2474

Max. :22.10 6: 296 6:2397 02 :1297 2014:2279

7:2112 (Other):7112 2015:1133

pickup\_hour dist

18 : 970 Min. : 0.000

19 : 967 1st Qu.: 1.257

20 : 936 Median : 2.172

21 : 894 Mean : 4.037

22 : 884 3rd Qu.: 3.905

12 : 780 Max. :5420.989

(Other):10229

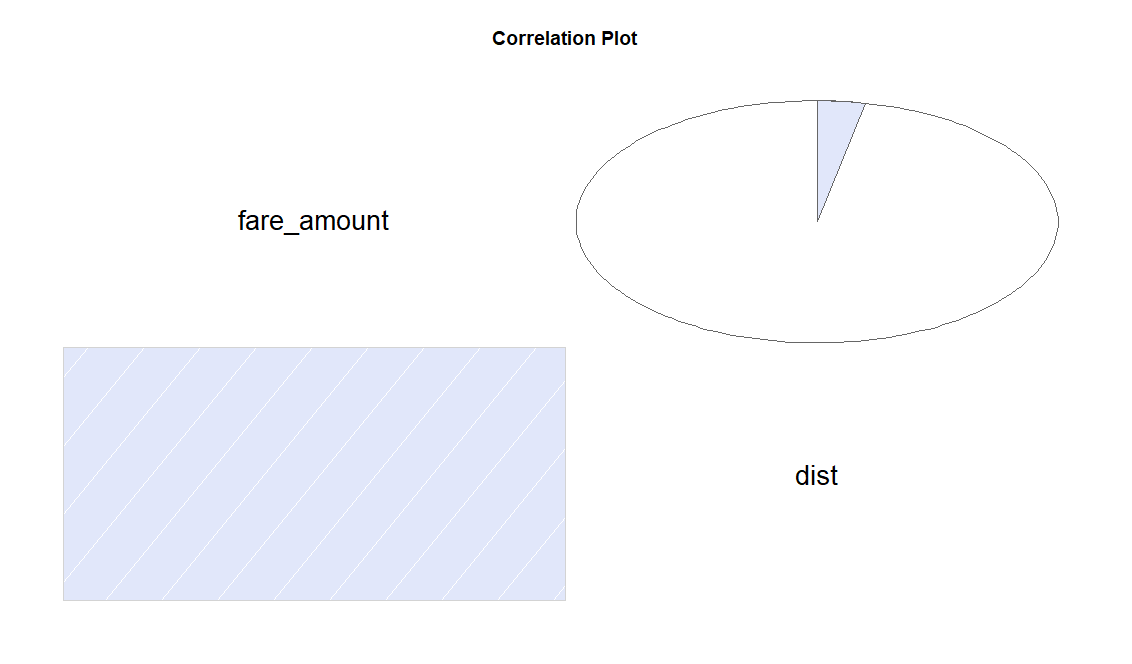
**Feature selection**

numeric\_index = sapply(train,is.numeric)

numeric\_data = train[,numeric\_index]

cnames = colnames(numeric\_data)

corrgram(train[,numeric\_index],upper.panel=panel.pie, main = "Correlation Plot")



aov\_results = aov(fare\_amount ~ passenger\_count + pickup\_hour + pickup\_weekday + pickup\_mnth + pickup\_yr,data = train)

summary(aov\_results)

Df Sum Sq Mean Sq F value Pr(>F)

passenger\_count 5 252 50.4 2.651 0.0212 \*

pickup\_hour 23 2552 111.0 5.840 < 2e-16 \*\*\*

pickup\_weekday 6 61 10.2 0.536 0.7814

pickup\_mnth 11 976 88.7 4.668 3.69e-07 \*\*\*

pickup\_yr 6 7111 1185.1 62.370 < 2e-16 \*\*\*

Residuals 15608 296573 19.0

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

train = subset(train,select=-pickup\_weekday)

test = subset(test,select=-pickup\_weekday)

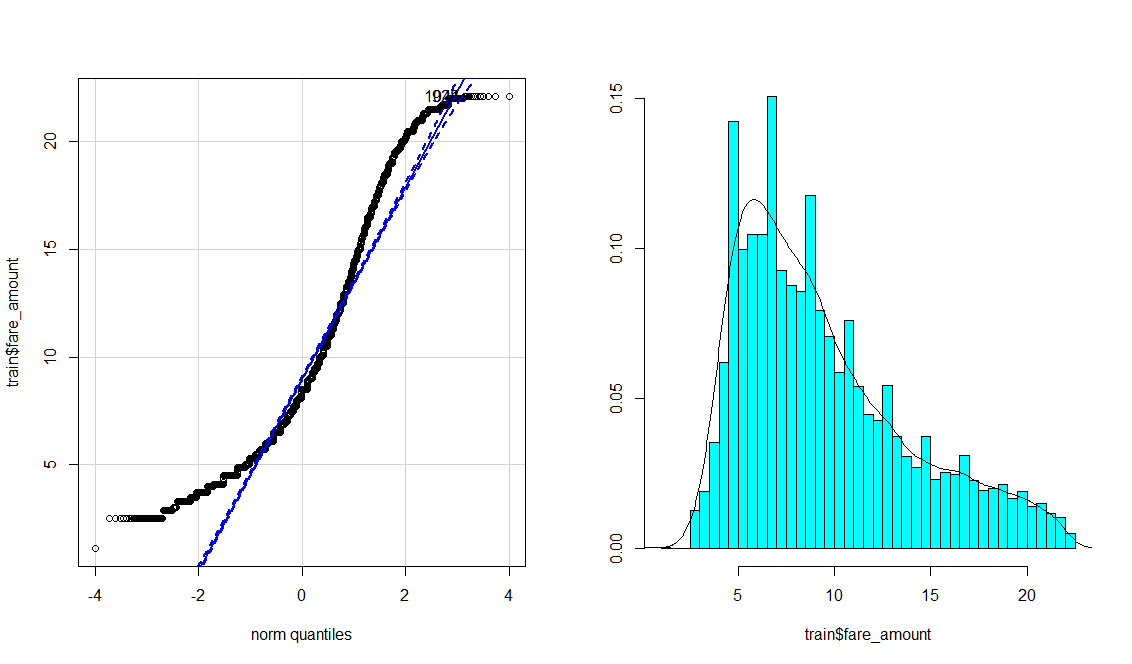
**Feature Scaling**

library(car)

par(mfrow=c(1,2))

qqPlot(train$fare\_amount) # qqPlot, it has a x values derived from gaussian distribution, if data is distributed normally then the sorted data points should lie very close to the solid reference line

truehist(train$fare\_amount) # truehist() scales the counts to give an estimate of the probability density.

lines(density(train$fare\_amount))

print('dist')

[1] "dist"

train[,'dist'] = (train[,'dist'] - min(train[,'dist']))/

(max(train[,'dist'] - min(train[,'dist'])))

set.seed(1000)

tr.idx = createDataPartition(train$fare\_amount,p=0.75,list = FALSE) # 75% in trainin and 25% in Validation Datasets

train\_data = train[tr.idx,]

test\_data = train[-tr.idx,]

rmExcept(c("test","train","df",'df1','df2','df3','test\_data','train\_data','test\_pickup\_datetime'))

Removed the following objects:

aov\_results, cnames, deg\_to\_rad, getmode, haversine, i, missing\_val, numeric\_data, numeric\_index, pickup\_time, pl1, tr.idx, vals

**Linear regression**

lm\_model = lm(fare\_amount ~.,data=train\_data)

summary(lm\_model)

Call:

lm(formula = fare\_amount ~ ., data = train\_data)

Residuals:

Min 1Q Median 3Q Max

-12.450 -3.307 -1.018 2.480 13.963

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) 8.96559 0.26452 33.894 < 2e-16 \*\*\*

passenger\_count2 0.19039 0.11588 1.643 0.100414

passenger\_count3 -0.19976 0.20391 -0.980 0.327296

passenger\_count4 0.06890 0.28362 0.243 0.808074

passenger\_count5 -0.11174 0.16744 -0.667 0.504581

passenger\_count6 -0.02183 0.30418 -0.072 0.942783

pickup\_mnth02 0.22368 0.19291 1.159 0.246280

pickup\_mnth03 0.29593 0.18660 1.586 0.112778

pickup\_mnth04 0.49900 0.18871 2.644 0.008198 \*\*

pickup\_mnth05 0.60326 0.18568 3.249 0.001162 \*\*

pickup\_mnth06 0.27109 0.18697 1.450 0.147120

pickup\_mnth07 0.46438 0.19924 2.331 0.019786 \*

pickup\_mnth08 0.51611 0.20328 2.539 0.011130 \*

pickup\_mnth09 0.84734 0.19822 4.275 1.93e-05 \*\*\*

pickup\_mnth10 0.76353 0.19447 3.926 8.68e-05 \*\*\*

pickup\_mnth11 0.71596 0.19850 3.607 0.000311 \*\*\*

pickup\_mnth12 0.54426 0.19727 2.759 0.005809 \*\*

pickup\_yr2010 -0.06239 0.14413 -0.433 0.665132

pickup\_yr2011 0.21910 0.14510 1.510 0.131071

pickup\_yr2012 0.63557 0.14376 4.421 9.91e-06 \*\*\*

pickup\_yr2013 1.45876 0.14442 10.101 < 2e-16 \*\*\*

pickup\_yr2014 1.68995 0.14754 11.454 < 2e-16 \*\*\*

pickup\_yr2015 1.59531 0.18508 8.620 < 2e-16 \*\*\*

pickup\_hour01 0.15091 0.31460 0.480 0.631456

pickup\_hour02 0.06142 0.35117 0.175 0.861166

pickup\_hour03 0.39361 0.37553 1.048 0.294597

pickup\_hour04 0.58770 0.40166 1.463 0.143448

pickup\_hour05 -0.84995 0.46672 -1.821 0.068618 .

pickup\_hour06 -1.68238 0.35238 -4.774 1.82e-06 \*\*\*

pickup\_hour07 -1.25367 0.28921 -4.335 1.47e-05 \*\*\*

pickup\_hour08 -1.06598 0.28054 -3.800 0.000146 \*\*\*

pickup\_hour09 -0.74709 0.27676 -2.699 0.006957 \*\*

pickup\_hour10 -1.24983 0.28473 -4.390 1.15e-05 \*\*\*

pickup\_hour11 -0.95906 0.27980 -3.428 0.000611 \*\*\*

pickup\_hour12 -0.94555 0.27707 -3.413 0.000646 \*\*\*

pickup\_hour13 -1.00153 0.27741 -3.610 0.000307 \*\*\*

pickup\_hour14 -0.61917 0.27525 -2.249 0.024501 \*

pickup\_hour15 -0.70365 0.28364 -2.481 0.013122 \*

pickup\_hour16 -0.94971 0.28434 -3.340 0.000840 \*\*\*

pickup\_hour17 -0.88904 0.27614 -3.220 0.001287 \*\*

pickup\_hour18 -0.70668 0.26349 -2.682 0.007328 \*\*

pickup\_hour19 -1.20546 0.26453 -4.557 5.24e-06 \*\*\*

pickup\_hour20 -0.55598 0.26497 -2.098 0.035904 \*

pickup\_hour21 -0.93603 0.26837 -3.488 0.000489 \*\*\*

pickup\_hour22 -0.68365 0.26857 -2.546 0.010924 \*

pickup\_hour23 -0.69455 0.27861 -2.493 0.012684 \*

dist 8.66311 3.37158 2.569 0.010198 \*

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.356 on 11699 degrees of freedom

Multiple R-squared: 0.03728, Adjusted R-squared: 0.03349

F-statistic: 9.848 on 46 and 11699 DF, p-value: < 2.2e-16

str(train\_data)

'data.frame': 11746 obs. of 6 variables:

$ fare\_amount : num 4.5 16.9 5.7 7.7 5.3 ...

$ passenger\_count: Factor w/ 6 levels "1","2","3","4",..: 1 1 2 1 1 1 1 1 2 1 ...

$ pickup\_mnth : Factor w/ 12 levels "01","02","03",..: 6 1 8 4 3 11 1 12 9 4 ...

$ pickup\_yr : Factor w/ 7 levels "2009","2010",..: 1 2 3 4 2 4 4 4 1 4 ...

$ pickup\_hour : Factor w/ 24 levels "00","01","02",..: 18 17 1 5 8 21 18 14 2 8 ...

$ dist : num 0.00019 0.001559 0.000256 0.000516 0.000369 ...

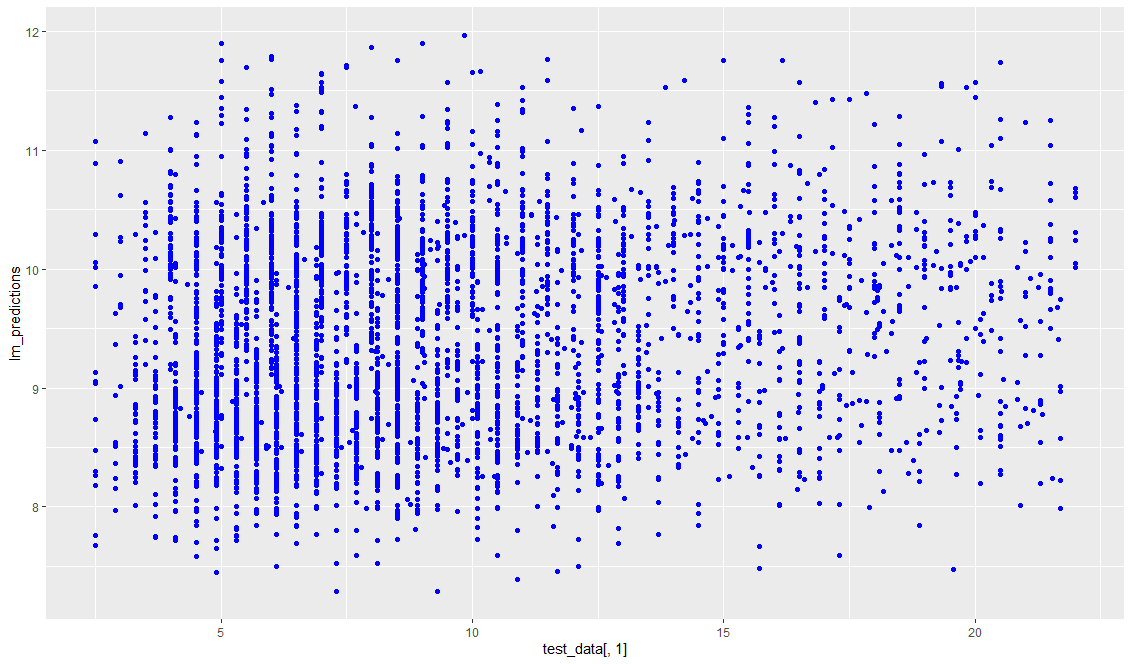
plot(lm\_model$fitted.values,rstandard(lm\_model),main = "Residual plot",

xlab = "Predicted values of fare\_amount",

ylab = "standardized residuals")

lm\_predictions = predict(lm\_model,test\_data[,2:6])

qplot(x = test\_data[,1], y = lm\_predictions, data = test\_data, color = I("blue"), geom = "point")



regr.eval(test\_data[,1],lm\_predictions)

mae mse rmse mape

3.5022600 19.1274501 4.3734940 0.4503374

**Decision Tree**

Dt\_model = rpart(fare\_amount ~ ., data = train\_data, method = "anova")

summary(Dt\_model)

Call:

rpart(formula = fare\_amount ~ ., data = train\_data, method = "anova")

n= 11746

CP nsplit rel error xerror xstd

1 0.51911212 0 1.0000000 1.0001854 0.013080481

2 0.07800495 1 0.4808879 0.4861941 0.007471865

3 0.03743835 2 0.4028829 0.4125529 0.007034712

4 0.01097480 3 0.3654446 0.3709656 0.007019269

5 0.01000000 4 0.3544698 0.3617584 0.007067199

Variable importance

dist

100

Node number 1: 11746 observations, complexity param=0.5191121

mean=9.395288, MSE=19.62856

left son=2 (8163 obs) right son=3 (3583 obs)

Primary splits:

dist < 0.0006177582 to the left, improve=0.5191121000, (0 missing)

pickup\_yr splits as LLLLRRR, improve=0.0215262400, (0 missing)

pickup\_hour splits as RRRRRLLLLLLLLLLLLLLLLLLL, improve=0.0065570310, (0 missing)

pickup\_mnth splits as LLLRRLLLRRRL, improve=0.0015229280, (0 missing)

passenger\_count splits as LRLRLR, improve=0.0004870972, (0 missing)

Node number 2: 8163 observations, complexity param=0.07800495

mean=7.280468, MSE=7.199322

left son=4 (4890 obs) right son=5 (3273 obs)

Primary splits:

dist < 0.0003348455 to the left, improve=0.306026700, (0 missing)

pickup\_yr splits as LLLLRRR, improve=0.039152040, (0 missing)

pickup\_hour splits as RRRRRLLRRRRRRRRRRRRRRRRR, improve=0.007024418, (0 missing)

pickup\_mnth splits as LLLLRLLLRLRL, improve=0.002056242, (0 missing)

passenger\_count splits as LLLRLR, improve=0.001741723, (0 missing)

Node number 3: 3583 observations, complexity param=0.03743835

mean=14.21339, MSE=14.54202

left son=6 (1584 obs) right son=7 (1999 obs)

Primary splits:

dist < 0.0009420487 to the left, improve=0.165662300, (0 missing)

pickup\_yr splits as LLLLRRR, improve=0.033294520, (0 missing)

pickup\_hour splits as RRRRRLLLRRLRRRRRLRRLRLRL, improve=0.007663497, (0 missing)

pickup\_mnth splits as RLRRRRLLRRRR, improve=0.004882979, (0 missing)

passenger\_count splits as RRLLLR, improve=0.004298045, (0 missing)

Surrogate splits:

pickup\_hour splits as RRRRRRRRRLRRRRRRRRLLRRRR, agree=0.57, adj=0.027, (0 split)

passenger\_count splits as RRRRRL, agree=0.56, adj=0.004, (0 split)

Node number 4: 4890 observations, complexity param=0.0109748

mean=6.066118, MSE=4.28564

left son=8 (2886 obs) right son=9 (2004 obs)

Primary splits:

dist < 0.0002282825 to the left, improve=0.120739800, (0 missing)

pickup\_yr splits as LLLLRRR, improve=0.035116790, (0 missing)

pickup\_hour splits as LLLLLLLLRRRRRRRRRRRRRLRL, improve=0.010808450, (0 missing)

pickup\_mnth splits as LLLLRLRRRLRR, improve=0.002831346, (0 missing)

passenger\_count splits as LLLRLR, improve=0.001251905, (0 missing)

Node number 5: 3273 observations

mean=9.094759, MSE=6.057651

Node number 6: 1584 observations

mean=12.46977, MSE=8.592556

Node number 7: 1999 observations

mean=15.59504, MSE=14.93835

Node number 8: 2886 observations

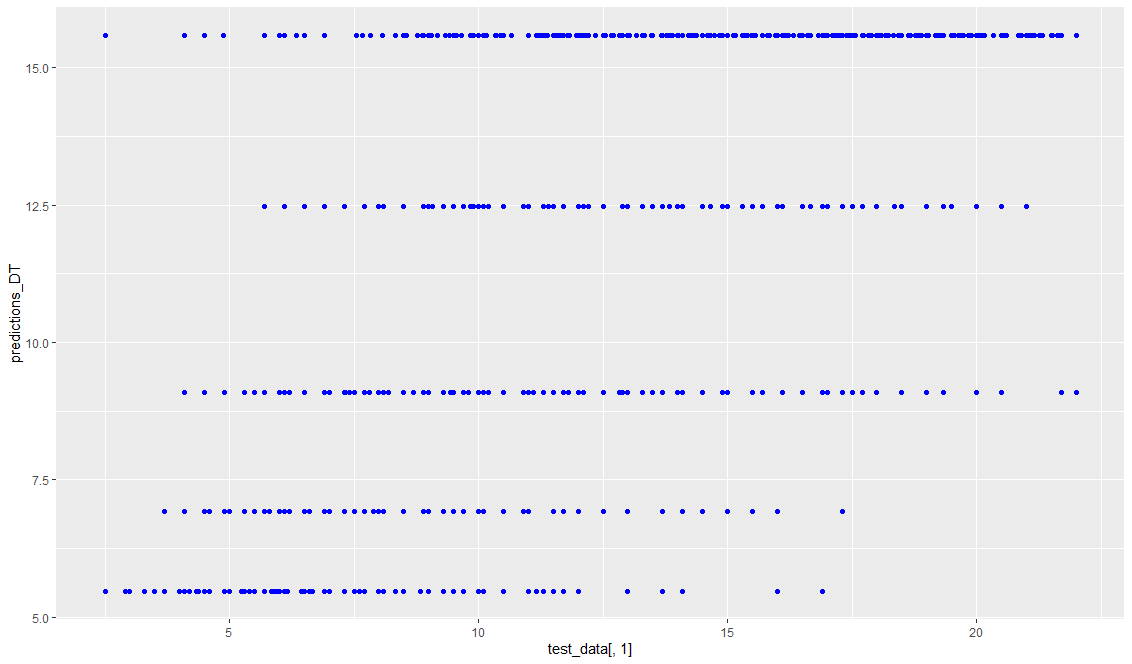
mean=5.466694, MSE=3.994985

Node number 9: 2004 observations

mean=6.92936, MSE=3.441584

predictions\_DT = predict(Dt\_model, test\_data[,2:6])

qplot(x = test\_data[,1], y = predictions\_DT, data = test\_data, color = I("blue"), geom = "point")



regr.eval(test\_data[,1],predictions\_DT)

mae mse rmse mape

1.9001012 6.6918569 2.5868624 0.2192849

**Random forest**

rf\_model = randomForest(fare\_amount ~.,data=train\_data)

summary(rf\_model)

Length Class Mode

call 3 -none- call

type 1 -none- character

predicted 11746 -none- numeric

mse 500 -none- numeric

rsq 500 -none- numeric

oob.times 11746 -none- numeric

importance 5 -none- numeric

importanceSD 0 -none- NULL

localImportance 0 -none- NULL

proximity 0 -none- NULL

ntree 1 -none- numeric

mtry 1 -none- numeric

forest 11 -none- list

coefs 0 -none- NULL

y 11746 -none- numeric

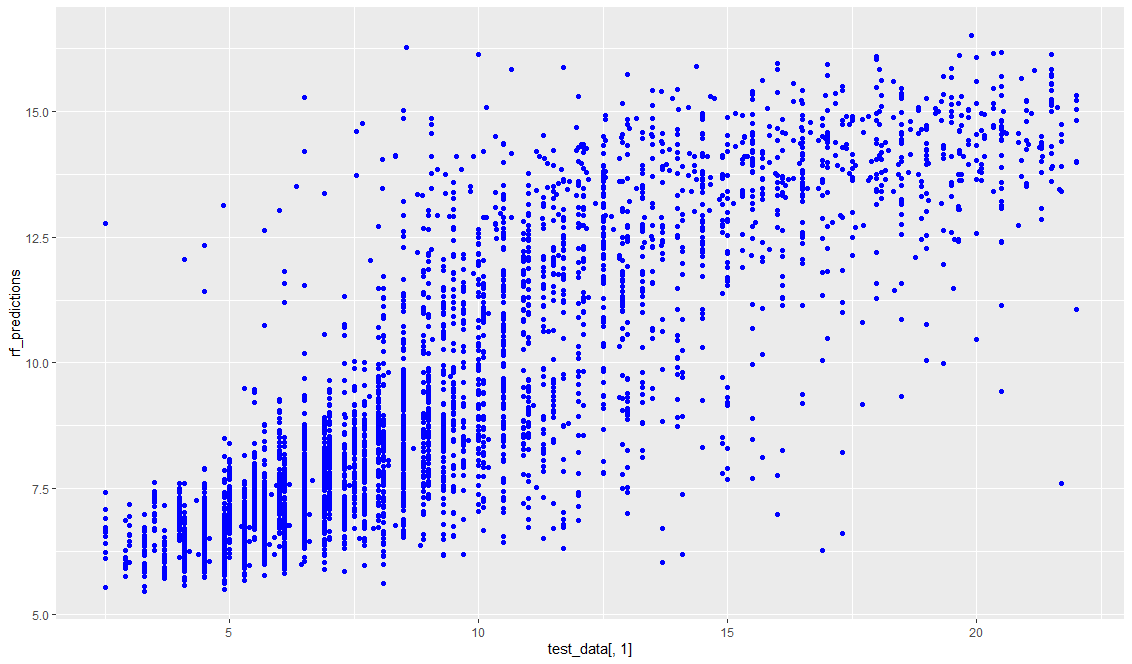
test 0 -none- NULL

inbag 0 -none- NULL

terms 3 terms call

rf\_predictions = predict(rf\_model,test\_data[,2:6])

qplot(x = test\_data[,1], y = rf\_predictions, data = test\_data, color = I("blue"), geom = "point")



regr.eval(test\_data[,1],rf\_predictions)

mae mse rmse mape

1.9103915 6.4214597 2.5340599 0.2342449

train\_data\_matrix = as.matrix(sapply(train\_data[-1],as.numeric))

test\_data\_data\_matrix = as.matrix(sapply(test\_data[-1],as.numeric))

xgboost\_model = xgboost(data = train\_data\_matrix,label = train\_data$fare\_amount,nrounds = 15,verbose = FALSE)

summary(xgboost\_model)

Length Class Mode

handle 1 xgb.Booster.handle externalptr

raw 57617 -none- raw

niter 1 -none- numeric

evaluation\_log 2 data.table list

call 13 -none- call

params 1 -none- list

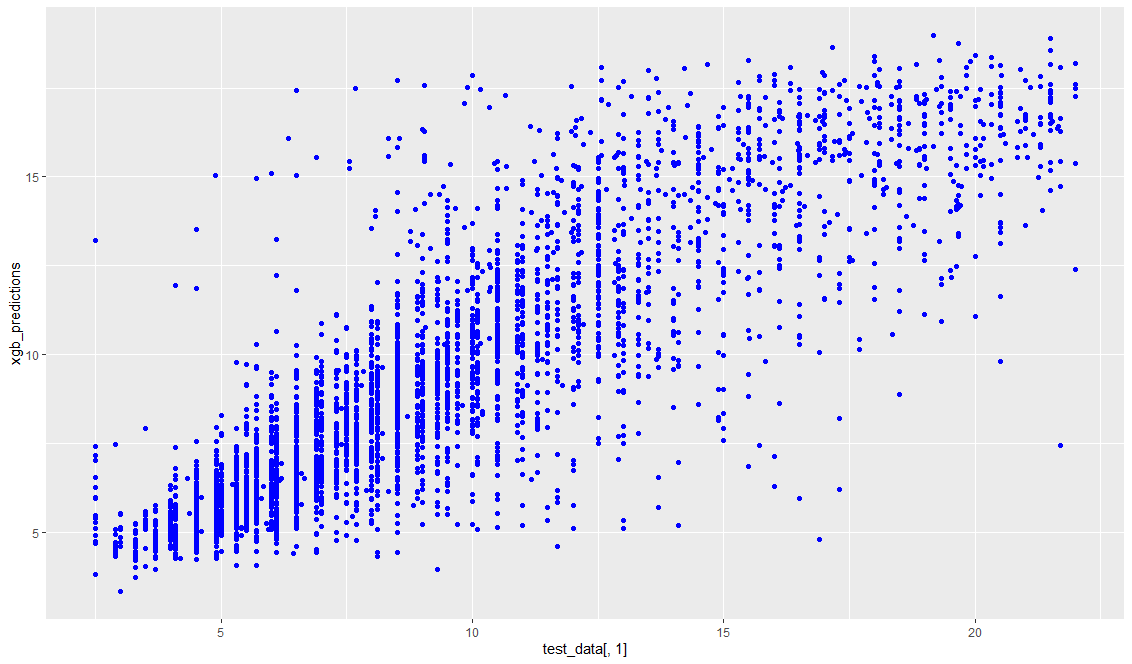
callbacks 1 -none- list

feature\_names 5 -none- character

nfeatures 1 -none- numeric

xgb\_predictions = predict(xgboost\_model,test\_data\_data\_matrix)

qplot(x = test\_data[,1], y = xgb\_predictions, data = test\_data, color = I("blue"), geom = "point")



regr.eval(test\_data[,1],xgb\_predictions)

|  |
| --- |
| mae mse rmse mape  1.6193378 5.2082001 2.2821481 0.1836901  train\_data\_matrix2 = as.matrix(sapply(train[-1],as.numeric))  test\_data\_matrix2 = as.matrix(sapply(test,as.numeric))  xgboost\_model2 = xgboost(data = train\_data\_matrix2,label = train$fare\_amount,nrounds = 15,verbose = FALSE) |
|  |
| |  | | --- | |  |   saveRDS(xgboost\_model2, "./final\_Xgboost\_model\_using\_R.rds")  super\_model <- readRDS("./final\_Xgboost\_model\_using\_R.rds")  print(super\_model)  ##### xgb.Booster  Handle is invalid! Suggest using xgb.Booster.complete  raw: 57.4 Kb  call:  xgb.train(params = params, data = dtrain, nrounds = nrounds,  watchlist = watchlist, verbose = verbose, print\_every\_n = print\_every\_n,  early\_stopping\_rounds = early\_stopping\_rounds, maximize = maximize,  save\_period = save\_period, save\_name = save\_name, xgb\_model = xgb\_model,  callbacks = callbacks)  params (as set within xgb.train):  silent = "1"  callbacks:  cb.evaluation.log()  # of features: 5  niter: 15  nfeatures : 5  evaluation\_log:  iter train\_rmse  1 7.170252  2 5.294364  ---  14 2.166089  15 2.163001  **Submission**  xgb = predict(super\_model,test\_data\_matrix2)  xgb\_pred = data.frame(test\_pickup\_datetime,"predictions" = xgb)  write.csv(xgb\_pred,"xgb\_predictions\_R.csv",row.names = FALSE) |