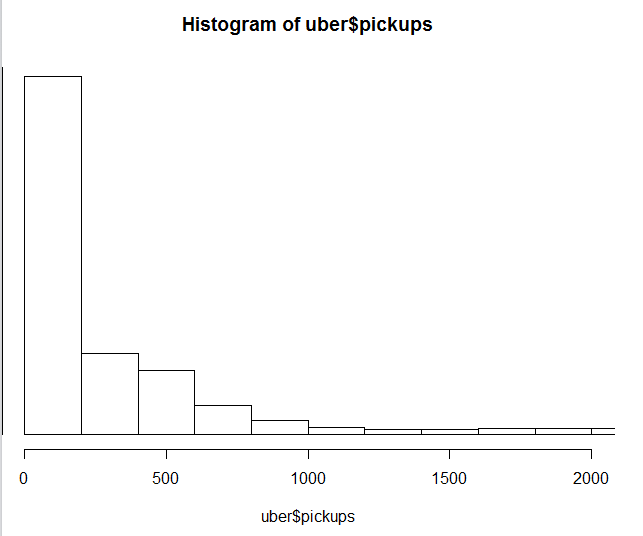
uber <- read.csv(file.choose()) #reading the dataset in R

|  |
| --- |
| > str(uber) #checking the classes of all the attributes  'data.frame': 26058 obs. of 12 variables:  $ pickup\_dt : Factor w/ 4343 levels "1/1/2015 1:00",..: 1 1 1 1 1 1 12 12 12 12 ...  $ Date : Factor w/ 181 levels "01/01/15","01/02/15",..: 1 1 1 1 1 1 1 1 1 1 ...  $ Time : Factor w/ 24 levels "1:00 AM","1:00 PM",..: 1 1 1 1 1 1 9 9 9 9 ...  $ borough : Factor w/ 6 levels "Bronx","Brooklyn",..: 1 2 3 4 5 6 1 2 3 4 ...  $ pickups : int 152 1519 0 5258 405 6 120 1229 0 4345 ...  $ Wind.Speed: num 5 5 5 5 5 5 3 3 3 3 ...  $ Visibility: num 10 10 10 10 10 10 10 10 10 10 ...  $ temp : num 30 30 30 30 30 30 30 30 30 30 ...  $ DewPoint : num 7 7 7 7 7 7 6 6 6 6 ...  $ Sea.Press : num 1024 1024 1024 1024 1024 ...  $ SnowDepth : num 0 0 0 0 0 0 0 0 0 0 ...  $ hday : Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 2 ...  > uber <- uber[!is.na(uber$borough),] #removing missing values  > uber1 <- uber #creating a backup  > uber1$hday <- as.numeric(uber1$hday) #converting to numeric for analysis  > uber1$borough <- as.numeric(uber1$borough) #converting to numeric for analysis  >  > describe(uber)#checking the attribute characterstics  vars n mean sd median trimmed mad min max range skew kurtosis se  pickup\_dt\* 1 26058 2172.00 1253.74 2172.0 2172.00 1610.10 1.0 4343.0 4342 0.00 -1.20 7.77  Date\* 2 26058 91.02 52.24 91.0 91.02 66.72 1.0 181.0 180 0.00 -1.20 0.32  Time\* 3 26058 12.50 6.92 13.0 12.50 8.90 1.0 24.0 23 0.00 -1.20 0.04  borough\* 4 26058 3.50 1.71 3.5 3.50 2.22 1.0 6.0 5 0.00 -1.27 0.01  pickups 5 26058 547.22 1037.31 86.0 271.53 127.50 0.0 7883.0 7883 2.79 7.99 6.43  Wind.Speed 6 26058 6.00 3.71 6.0 5.85 2.97 0.0 21.0 21 0.42 0.42 0.02  Visibility 7 26058 8.82 2.44 10.0 9.47 0.00 0.0 10.0 10 -2.05 2.91 0.02  temp 8 26058 47.49 19.77 45.0 47.34 23.72 2.0 89.0 87 0.07 -1.03 0.12  DewPoint 9 26058 30.65 21.24 30.0 30.69 26.69 -16.0 73.0 89 0.03 -1.03 0.13  Sea.Press 10 26058 1017.81 7.78 1018.2 1017.79 7.86 991.4 1043.4 52 0.05 0.05 0.05  SnowDepth 11 26058 2.54 4.52 0.0 1.54 0.00 0.0 19.0 19 1.58 1.29 0.03  hday\* 12 26058 1.04 0.19 1.0 1.00 0.00 1.0 2.0 1 4.80 21.04 0.00 |
|  |
| |  | | --- | | > | |

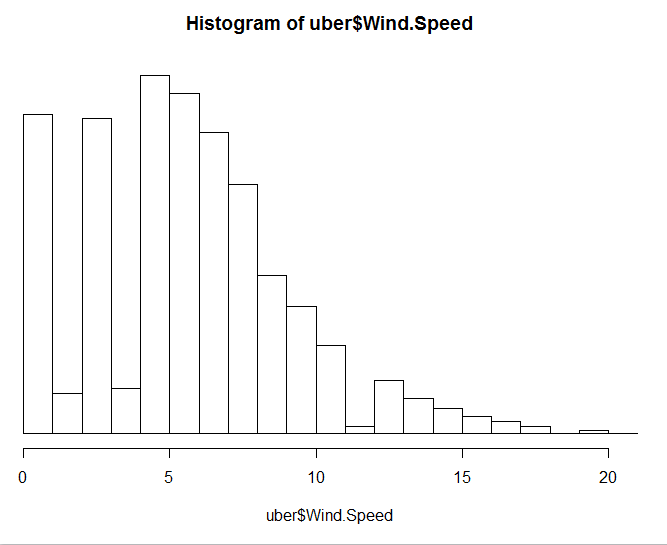
#checking attributed distributions

hist(uber$pickups,breaks = 30,xlim = c(0,2000))

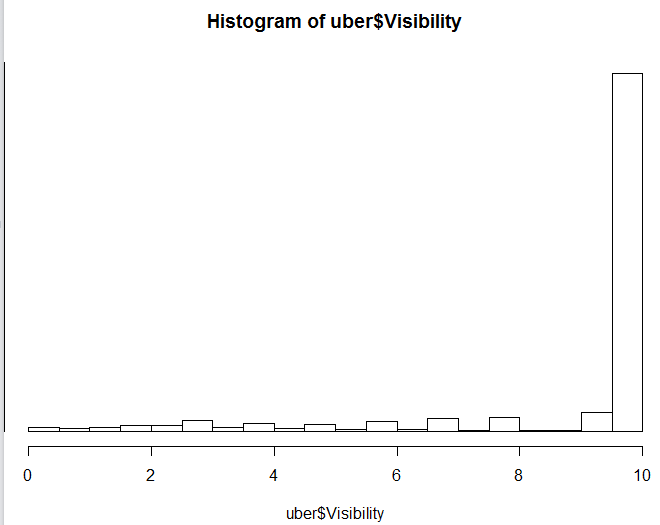


|  |
| --- |
| > t <- aggregate(uber$pickups,by=list(uber$Time), sum) #contigency tables for time versus pickups  > t  Group.1 x  1 1:00 AM 491828  2 1:00 PM 530195  3 10:00 AM 550679  4 10:00 PM 927836  5 11:00 AM 509621  6 11:00 PM 886636  7 12:00 AM 706428  8 12:00 PM 532132  9 2:00 AM 319332  10 2:00 PM 557528  11 3:00 AM 213874  12 3:00 PM 618418  13 4:00 AM 173531  14 4:00 PM 683643  15 5:00 AM 174210  16 5:00 PM 801745  17 6:00 AM 230915  18 6:00 PM 931751  19 7:00 AM 359688  20 7:00 PM 1012936  21 8:00 AM 525289  22 8:00 PM 977314  23 9:00 AM 607574  24 9:00 PM 936410  > prop.table(t[,2]) #checking propotions of pickups on wach time  [1] 0.03449122 0.03718184 0.03861836 0.06506786 0.03573902 0.06217856 0.04954082 0.03731768 0.02239431 0.03909867 0.01499869 0.04336880  [13] 0.01216949 0.04794294 0.01221711 0.05622527 0.01619375 0.06534241 0.02522442 0.07103581 0.03683779 0.06853768 0.04260833 0.06566914 |
|  |
| |  | | --- | | > | |

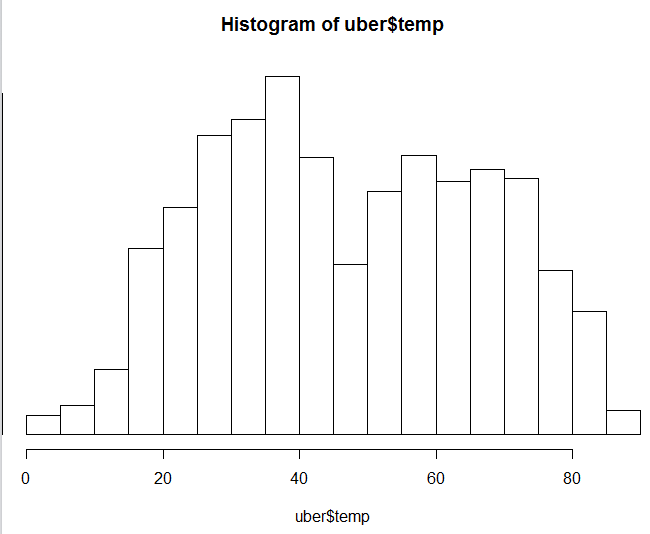
hist(uber$Wind.Speed) #histogram for the wind speed vattribute



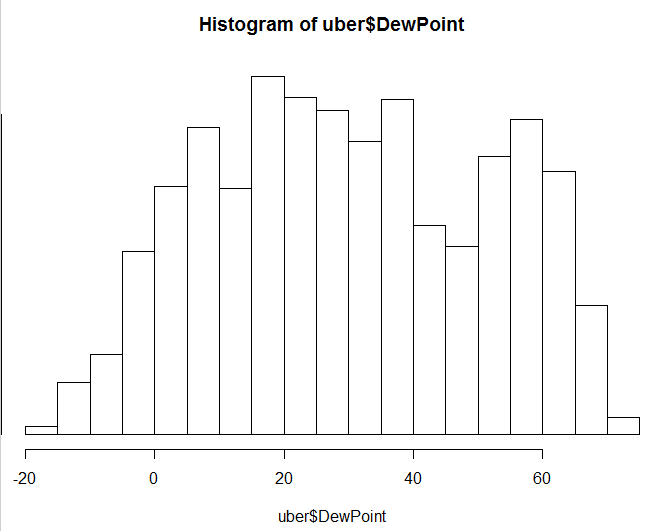
hist(uber$Visibility)#histogram for the visibility attribute



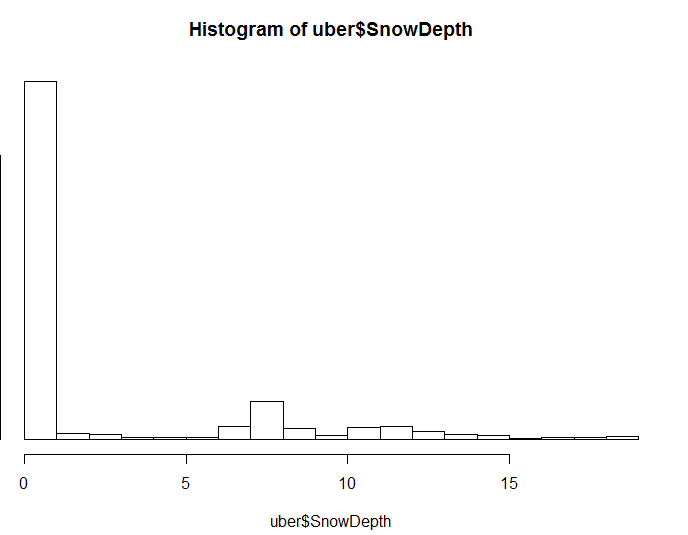
hist(uber$temp)#histogram for the temprature variable



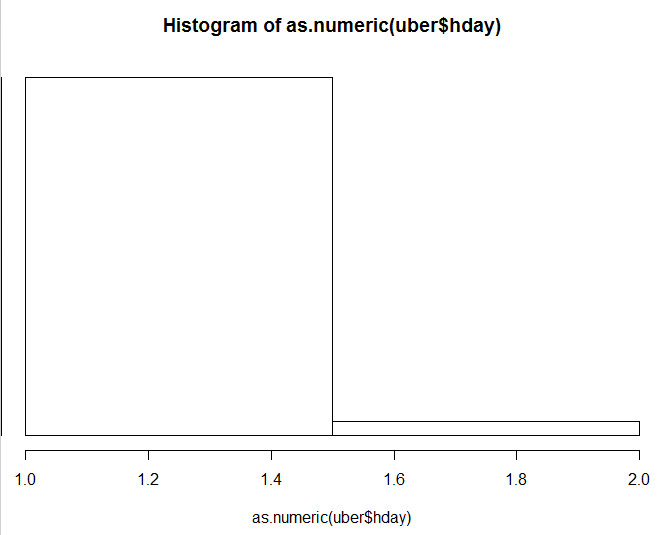
hist(uber$DewPoint)#histogram for the dewpoint variable



hist(uber$SnowDepth)#histogram for the snowdepth variable



hist(as.numeric(uber$hday))#histohram for the holiday variable



> hdaypk <- aggregate(uber$pickups,by=list(uber$hday),mean)#number of pickups aggregated by holidays

> hdaypk

Group.1 x

1 N 549.5552

2 Y 488.8792

> nhday <- table(uber$hday)#checking the number of holdays in last 6 months

> nhday

N Y

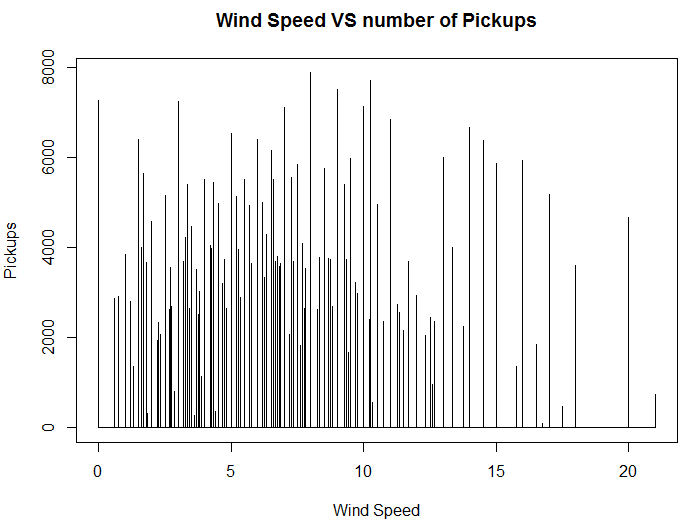
25056 1002

> prop.table(hdaypk[,2])

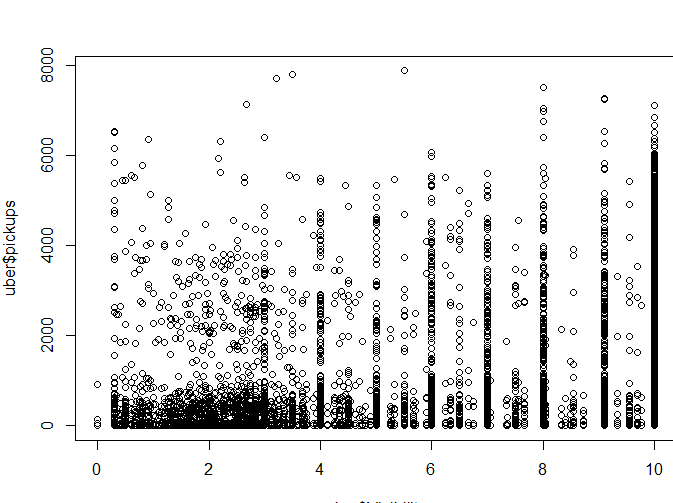
[1] 0.5292151 0.4707849

> prop.table(nhday)

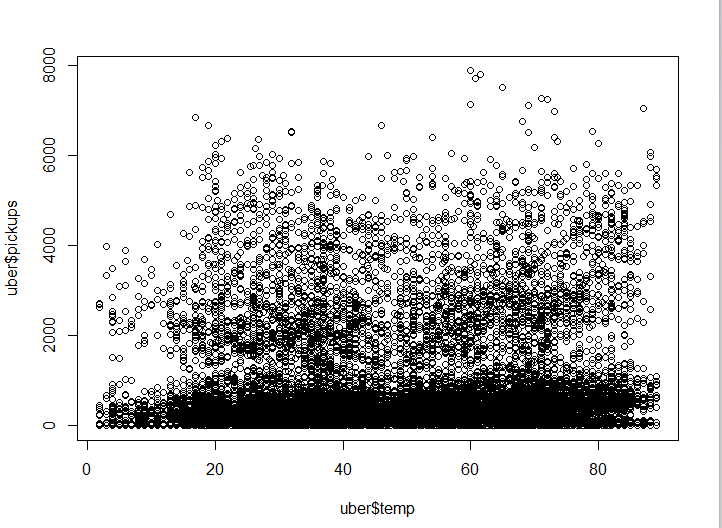
plot(uber$Wind.Speed,uber$pickups,type = 's')#Wind speed SV number of pickups



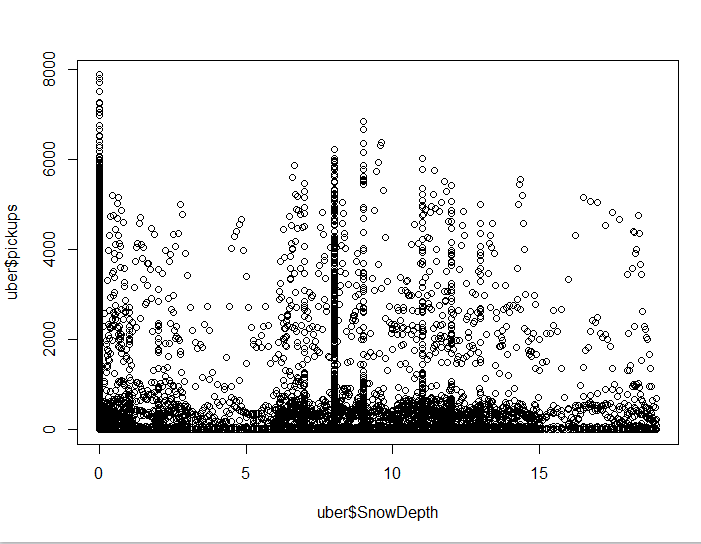
plot(uber$Visibility,uber$pickups)# visibility VS number of pickups



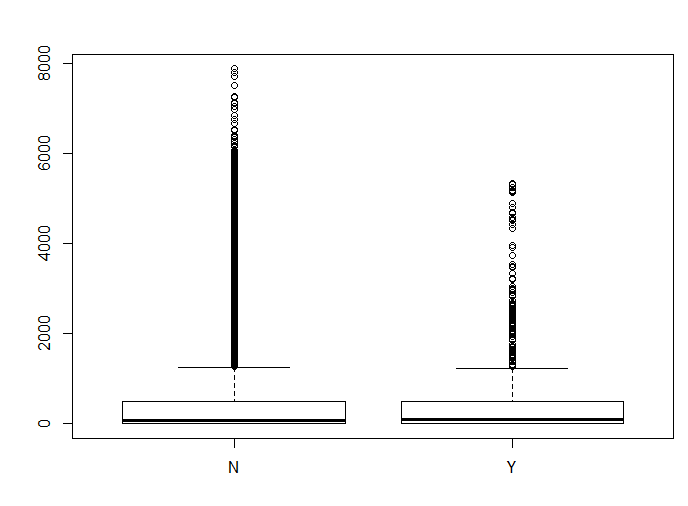
plot(uber$temp,uber$pickups)# temprature versus the number of pickups



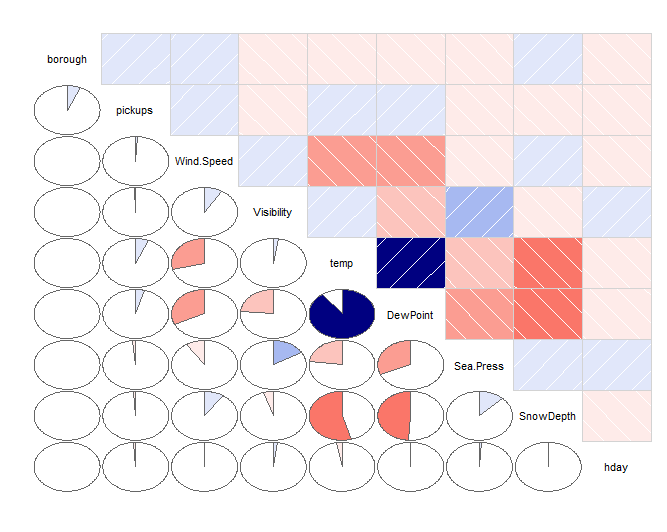
plot(uber$SnowDepth,uber$pickups)#snowdepthe versus the number of pickups



plot(uber$hday,uber$pickups)#holiday vwesus the number of pickups



corrgram(uber1,lower.panel = panel.pie)#corrogram for the variables



> cor(uber1[,4:12]) #correlation between all the attributes

borough pickups Wind.Speed Visibility temp DewPoint

borough 1.00000000 0.068993734 0.000000000 0.000000000 0.00000000 0.000000000

pickups 0.06899373 1.000000000 0.009741013 -0.008428765 0.06369218 0.040081808

Wind.Speed 0.00000000 0.009741013 1.000000000 0.086177229 -0.29612612 -0.321606245

Visibility 0.00000000 -0.008428765 0.086177229 1.000000000 0.02521425 -0.231294340

temp 0.00000000 0.063692182 -0.296126122 0.025214254 1.00000000 0.896544483

DewPoint 0.00000000 0.040081808 -0.321606245 -0.231294340 0.89654448 1.000000000

Sea.Press 0.00000000 -0.015707747 -0.092761122 0.167039095 -0.22453670 -0.311156018

SnowDepth 0.00000000 -0.009675919 0.097040543 -0.047833727 -0.54555764 -0.489371808

hday 0.00000000 -0.011247766 -0.006356239 0.022061217 -0.02776446 -0.007857157

Sea.Press SnowDepth hday

borough 0.000000000 0.000000000 0.000000000

pickups -0.015707747 -0.009675919 -0.011247766

Wind.Speed -0.092761122 0.097040543 -0.006356239

Visibility 0.167039095 -0.047833727 0.022061217

temp -0.224536701 -0.545557643 -0.027764464

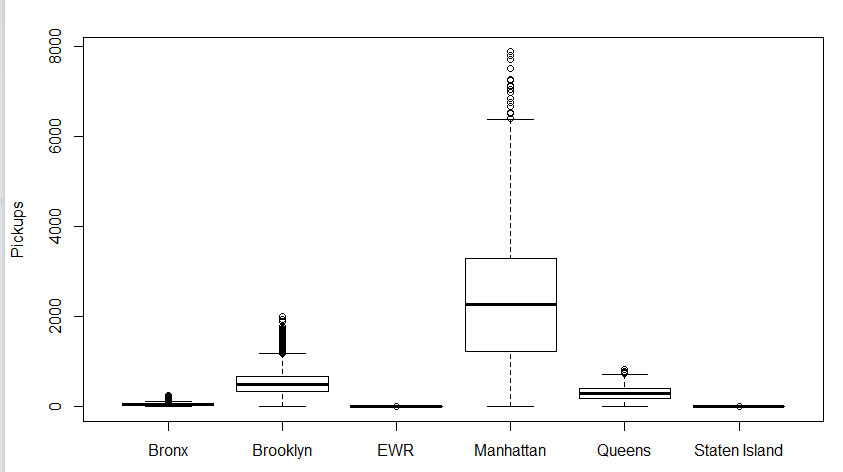
DewPoint -0.311156018 -0.489371808 -0.007857157

Sea.Press 1.000000000 0.121507627 0.009132828

SnowDepth 0.121507627 1.000000000 -0.006448875

hday 0.009132828 -0.006448875 1.000000000

plot(uber$borough,uber1$pickups)#pickup plave VS number of pickups



|  |
| --- |
| > aggregate(uber1$pickups,by=list(uber$borough),sum) #sum of the pickups from each spot  Group.1 x  1 Bronx 220047  2 Brooklyn 2321035  3 EWR 105  4 Manhattan 10367841  5 Queens 1343528  6 Staten Island 6957 |
|  |
| |  | | --- | | > | |

> #Our main objective here was to check what is the effect of various environment factors on Uber Pickups

> #thus our target variable is "Number of Pickups"

> #We run Leniar Regression first by considering all the variables and the removing all the variables that are least significant one by one

>

> training <- uber1

> #first regression

> reg1 <- lm(training$pickups~training$borough+training$Wind.Speed+training$Visibility+training$temp+training$DewPoint+training$Sea.Press+training$SnowDepth+training$hday,data = training)

> summary(reg1)

Call:

lm(formula = training$pickups ~ training$borough + training$Wind.Speed +

training$Visibility + training$temp + training$DewPoint +

training$Sea.Press + training$SnowDepth + training$hday,

data = training)

Residuals:

Min 1Q Median 3Q Max

-997.9 -528.9 -354.9 -22.9 7286.7

Coefficients:

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

(Intercept) 787.5056 912.5316 0.863 0.388

training$borough 41.9050 3.7378 11.211 < 2e-16 \*\*\*

training$Wind.Speed 8.1671 1.8700 4.367 1.26e-05 \*\*\*

training$Visibility -19.6788 3.2032 -6.143 8.19e-10 \*\*\*

training$temp 11.4272 0.8917 12.815 < 2e-16 \*\*\*

training$DewPoint -6.8535 0.8535 -8.030 1.01e-15 \*\*\*

training$Sea.Press -0.5785 0.8920 -0.649 0.517

training$SnowDepth 8.2430 1.6933 4.868 1.13e-06 \*\*\*

training$hday -26.0215 33.2986 -0.781 0.435

---

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

Residual standard error: 1030 on 26049 degrees of freedom

Multiple R-squared: 0.01346, Adjusted R-squared: 0.01315

F-statistic: 44.42 on 8 and 26049 DF, p-value: < 2.2e-16

> #second regresson

> reg2 <- lm(training$pickups~training$borough+training$Wind.Speed+training$Visibility+training$temp+training$DewPoint+training$Sea.Press+training$SnowDepth,data = training)

> summary(reg2)

Call:

lm(formula = training$pickups ~ training$borough + training$Wind.Speed +

training$Visibility + training$temp + training$DewPoint +

training$Sea.Press + training$SnowDepth, data = training)

Residuals:

Min 1Q Median 3Q Max

-998.4 -529.0 -354.9 -23.8 7287.7

Coefficients:

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

(Intercept) 766.2180 912.1181 0.840 0.401

training$borough 41.9050 3.7378 11.211 < 2e-16 \*\*\*

training$Wind.Speed 8.1840 1.8699 4.377 1.21e-05 \*\*\*

training$Visibility -19.8126 3.1986 -6.194 5.95e-10 \*\*\*

training$temp 11.4782 0.8893 12.906 < 2e-16 \*\*\*

training$DewPoint -6.8942 0.8519 -8.093 6.07e-16 \*\*\*

training$Sea.Press -0.5844 0.8920 -0.655 0.512

training$SnowDepth 8.2747 1.6928 4.888 1.02e-06 \*\*\*

---

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

Residual standard error: 1030 on 26050 degrees of freedom

Multiple R-squared: 0.01343, Adjusted R-squared: 0.01317

F-statistic: 50.68 on 7 and 26050 DF, p-value: < 2.2e-16

> #third regression and the best model obtained

> reg3 <- lm(training$pickups~training$borough+training$Wind.Speed+training$Visibility+training$temp+training$DewPoint+training$SnowDepth,data = training)

> summary(reg3)

Call:

lm(formula = training$pickups ~ training$borough + training$Wind.Speed +

training$Visibility + training$temp + training$DewPoint +

training$SnowDepth, data = training)

Residuals:

Min 1Q Median 3Q Max

-1001.0 -529.6 -355.0 -24.0 7284.9

Coefficients:

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

(Intercept) 169.1823 37.8555 4.469 7.89e-06 \*\*\*

training$borough 41.9050 3.7377 11.211 < 2e-16 \*\*\*

training$Wind.Speed 8.4482 1.8258 4.627 3.73e-06 \*\*\*

training$Visibility -19.9081 3.1952 -6.231 4.72e-10 \*\*\*

training$temp 11.4328 0.8866 12.895 < 2e-16 \*\*\*

training$DewPoint -6.7753 0.8323 -8.140 4.12e-16 \*\*\*

training$SnowDepth 8.2937 1.6925 4.900 9.63e-07 \*\*\*

---

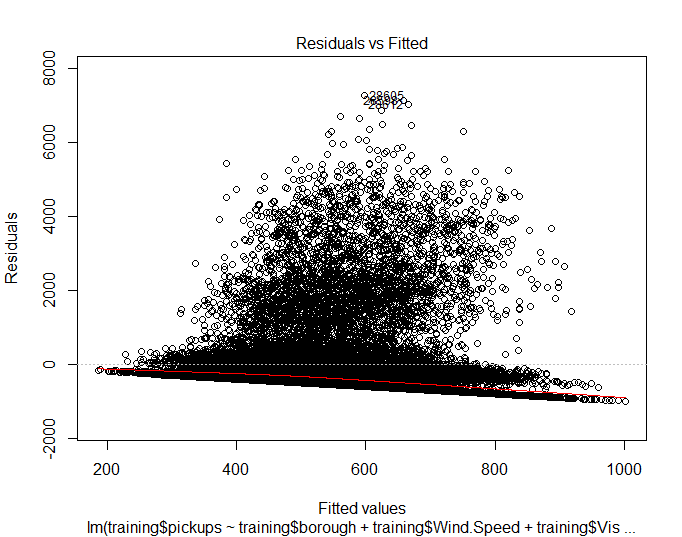
Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1030 on 26051 degrees of freedom

Multiple R-squared: 0.01342, Adjusted R-squared: 0.01319

F-statistic: 59.05 on 6 and 26051 DF, p-value: < 2.2e-16

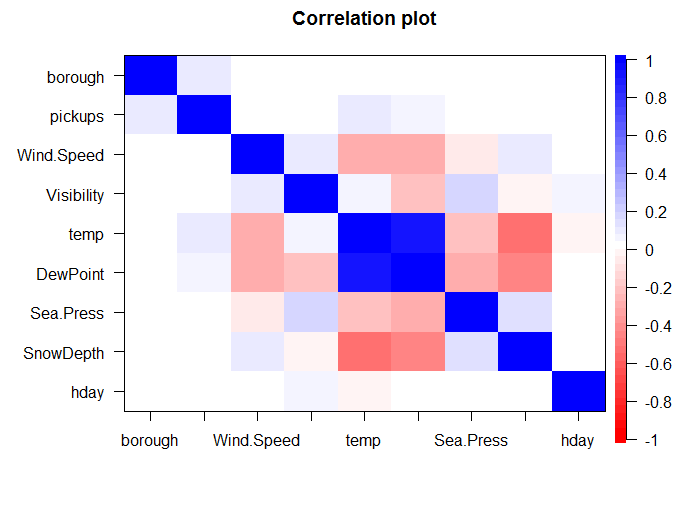
> plot(reg3)



#checking correlation between the variables that have a significant effect on our target variable i.e. "Number of Pickups"

subset.uber1 <- uber1[,4:12]

cor.plot(subset.uber1)#correlation plot



> cov2cor(cov(subset.uber1)) #covariance

borough pickups Wind.Speed Visibility temp DewPoint Sea.Press

borough 1.00000000 0.068993734 0.000000000 0.000000000 0.00000000 0.000000000 0.000000000

pickups 0.06899373 1.000000000 0.009741013 -0.008428765 0.06369218 0.040081808 -0.015707747

Wind.Speed 0.00000000 0.009741013 1.000000000 0.086177229 -0.29612612 -0.321606245 -0.092761122

Visibility 0.00000000 -0.008428765 0.086177229 1.000000000 0.02521425 -0.231294340 0.167039095

temp 0.00000000 0.063692182 -0.296126122 0.025214254 1.00000000 0.896544483 -0.224536701

DewPoint 0.00000000 0.040081808 -0.321606245 -0.231294340 0.89654448 1.000000000 -0.311156018

Sea.Press 0.00000000 -0.015707747 -0.092761122 0.167039095 -0.22453670 -0.311156018 1.000000000

SnowDepth 0.00000000 -0.009675919 0.097040543 -0.047833727 -0.54555764 -0.489371808 0.121507627

hday 0.00000000 -0.011247766 -0.006356239 0.022061217 -0.02776446 -0.007857157 0.009132828

SnowDepth hday

borough 0.000000000 0.000000000

pickups -0.009675919 -0.011247766

Wind.Speed 0.097040543 -0.006356239

Visibility -0.047833727 0.022061217

temp -0.545557643 -0.027764464

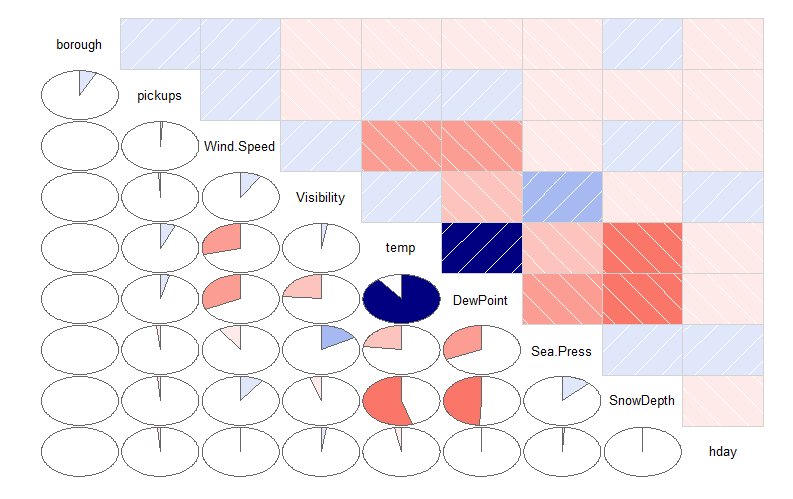
DewPoint -0.489371808 -0.007857157

Sea.Press 0.121507627 0.009132828

SnowDepth 1.000000000 -0.006448875

hday -0.006448875 1.000000000

corrgram(subset.uber1,lower.panel = panel.pie)#corrogram

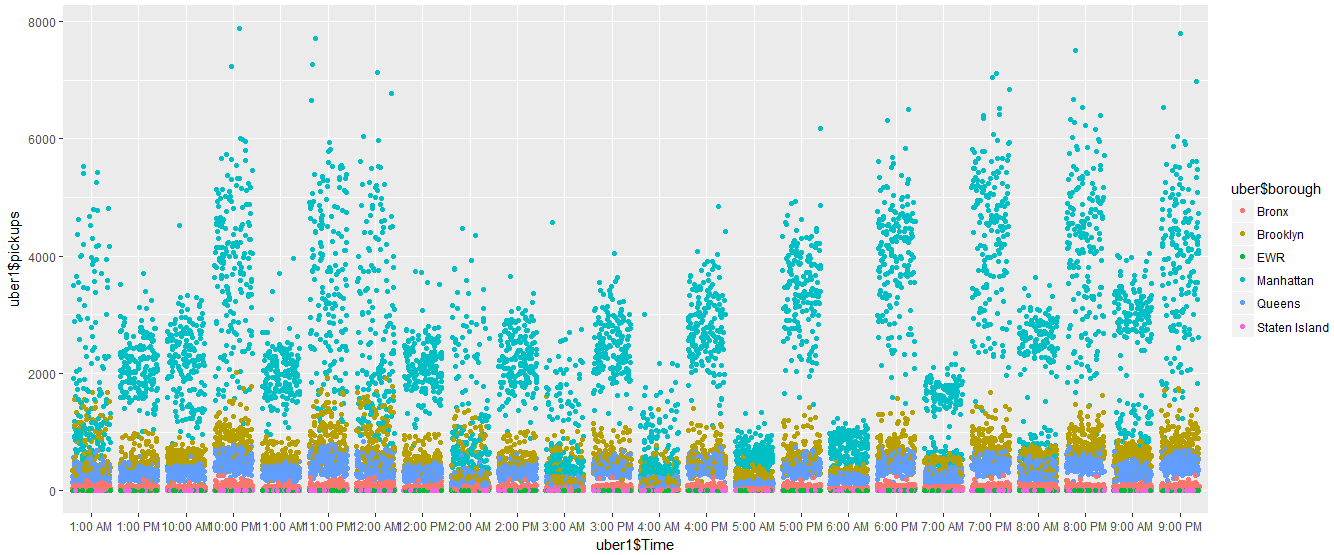


#pplotting Time VS number of pickups by highlighting the pickup place

#so that we get a better ideas as to where the pickups happern the most at what time

pl.pik <- ggplot(data = uber1,aes(x=uber1$Time,y=uber1$pickups))

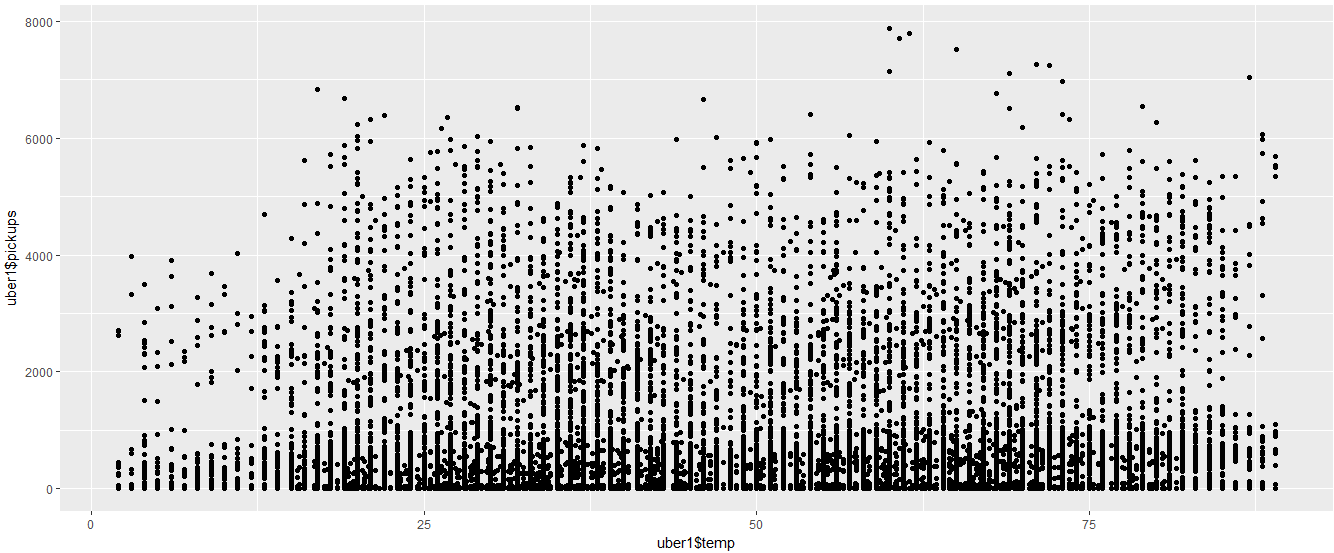
pl.pik+geom\_jitter(aes(colour= uber$borough))#greatjob



#plotting temprature VS number of pickups

temp.pick <- ggplot(data = uber1,aes(x=uber1$temp,y=uber1$pickups))

temp.pick+geom\_point()



> aggregate(uber1$pickups,by=list(uber1$temp),sum)

Group.1 x

1 2.00000 10123

2 3.00000 9287

3 4.00000 33462

4 5.00000 13204

5 6.00000 21389

6 7.00000 10568

7 8.00000 18711

8 9.00000 28085

9 10.00000 17725

10 11.00000 17840

11 12.00000 14171

12 13.00000 62276

13 14.00000 45663

14 15.00000 73941

15 15.50000 2767

16 15.66667 4650

17 16.00000 47762

18 16.75000 4008

19 17.00000 116214

20 17.50000 2931

21 17.75000 3117

22 18.00000 118589

23 18.50000 682

24 19.00000 139459

25 19.25000 3445

26 19.33333 2370

27 19.50000 625

28 19.66667 2249

29 19.75000 4396

30 20.00000 239729

31 20.33333 5875

32 20.50000 3123

33 20.80000 4387

34 21.00000 175606

35 21.33333 5496

36 21.66667 32

37 22.00000 153162

38 22.50000 1706

39 23.00000 144307

40 23.25000 4176

41 23.50000 4464

42 23.66667 2973

43 23.80000 2406

44 24.00000 167308

45 24.50000 5531

46 24.66667 840

47 25.00000 151594

48 25.33333 0

49 25.50000 15529

50 25.75000 3153

51 26.00000 230934

52 26.25000 3290

53 26.33333 7568

54 26.50000 6419

55 26.66667 1853

56 26.75000 7911

57 27.00000 206770

58 27.20000 4426

59 27.33333 6931

60 27.50000 3490

61 27.66667 1057

62 28.00000 172571

63 28.20000 2928

64 28.50000 3314

65 28.57143 1609

66 28.66667 5109

67 28.75000 3594

68 28.80000 3078

69 29.00000 232918

70 29.33333 3495

71 29.66667 2891

72 29.75000 4293

73 29.80000 3552

74 29.83333 4464

75 30.00000 236772

76 30.20000 4184

77 30.33333 2333

78 30.50000 3207

79 30.75000 6549

80 31.00000 189379

81 31.33333 2238

82 31.50000 16331

83 31.66667 2601

84 31.80000 2628

85 32.00000 186306

86 32.20000 2502

87 32.50000 4417

88 32.66667 405

89 33.00000 194356

90 33.16667 3176

91 33.20000 423

92 33.40000 2347

93 33.50000 4865

94 33.66667 8567

95 33.75000 885

96 34.00000 206008

97 34.25000 5817

98 34.33333 1228

99 34.50000 6087

100 35.00000 233689

101 35.20000 5323

102 35.33333 6659

103 35.60000 4804

104 35.75000 3937

105 36.00000 272766

106 36.33333 11016

107 36.40000 6384

108 36.50000 16584

109 36.60000 4918

110 36.66667 2580

111 36.75000 6539

112 37.00000 288379

113 37.25000 3875

114 37.57143 3297

115 37.60000 3694

116 37.66667 3663

117 37.75000 3065

118 38.00000 234502

119 38.33333 8998

120 38.50000 10732

121 39.00000 220222

122 39.16667 3485

123 39.33333 2707

124 39.50000 3159

125 39.60000 1810

126 39.66667 3864

127 39.80000 3493

128 40.00000 153063

129 40.25000 3104

130 40.33333 4019

131 41.00000 213455

132 41.25000 3335

133 41.33333 1706

134 41.66667 4013

135 41.75000 920

136 42.00000 214179

137 42.33333 5913

138 42.50000 21624

139 42.66667 712

140 42.83333 1297

141 43.00000 151159

142 43.75000 4699

143 44.00000 166764

144 44.20000 5092

145 44.25000 1361

146 44.33333 4921

147 44.50000 2606

148 44.60000 426

149 44.66667 403

150 45.00000 154390

151 45.50000 1129

152 45.66667 3614

153 45.75000 3978

154 46.00000 139320

155 46.33333 810

156 46.50000 7818

157 47.00000 97181

158 47.50000 5619

159 48.00000 95190

160 48.50000 6541

161 49.00000 134676

162 49.25000 3500

163 49.50000 8955

164 50.00000 176691

165 50.25000 5015

166 50.33333 2428

167 50.50000 4246

168 51.00000 186314

169 51.33333 5624

170 51.40000 3279

171 52.00000 170620

172 52.50000 7962

173 53.00000 142832

174 53.20000 2550

175 53.40000 610

176 53.50000 944

177 54.00000 191845

178 54.50000 3066

179 54.66667 10506

180 55.00000 143753

181 55.25000 3941

182 55.33333 11709

183 55.50000 7735

184 55.66667 7949

185 55.75000 4826

186 55.83333 4677

187 56.00000 221082

188 56.20000 4511

189 56.25000 3549

190 56.50000 10390

191 57.00000 178089

192 57.33333 2909

193 57.40000 5307

194 57.50000 18353

195 58.00000 151491

196 58.25000 4388

197 58.33333 861

198 58.50000 7349

199 59.00000 210886

200 59.25000 7756

201 59.33333 10460

202 59.50000 7241

203 59.66667 3969

204 59.75000 4454

205 60.00000 238320

206 60.33333 1592

207 60.42857 5890

208 60.50000 13472

209 60.75000 10510

210 61.00000 226239

211 61.16667 3316

212 61.50000 10315

213 61.66667 3516

214 62.00000 157274

215 62.40000 5053

216 62.50000 12830

217 62.66667 4116

218 63.00000 175774

219 63.33333 4638

220 63.50000 8398

221 64.00000 216442

222 64.50000 18742

223 64.66667 10421

224 64.75000 4400

225 65.00000 172279

226 65.33333 4191

227 65.50000 5823

228 65.66667 6018

229 65.80000 4502

230 66.00000 172574

231 66.20000 2077

232 66.25000 728

233 66.50000 8952

234 66.75000 3593

235 66.80000 3259

236 66.83333 1340

237 66.85714 2247

238 67.00000 219900

239 67.25000 4136

240 67.50000 9373

241 67.80000 4476

242 68.00000 254348

243 68.33333 4766

244 68.50000 21167

245 68.66667 4747

246 68.75000 1746

247 69.00000 260701

248 69.33333 13326

249 69.50000 12840

250 69.66667 5870

251 69.75000 3437

252 70.00000 205400

253 70.50000 3080

254 70.66667 9235

255 70.80000 553

256 71.00000 249203

257 71.25000 2764

258 71.50000 15092

259 72.00000 204297

260 72.33333 1113

261 72.66667 1141

262 73.00000 232716

263 73.50000 15020

264 73.66667 12976

265 73.75000 4190

266 74.00000 140462

267 74.28571 4586

268 74.50000 16380

269 75.00000 157466

270 75.33333 6864

271 75.66667 3416

272 76.00000 180016

273 76.66667 5919

274 77.00000 188684

275 77.50000 4113

276 77.66667 4676

277 78.00000 137951

278 78.25000 926

279 78.50000 11081

280 79.00000 161157

281 79.50000 10721

282 79.75000 3881

283 80.00000 147256

284 80.50000 8973

285 80.66667 5194

286 81.00000 135522

287 81.50000 4895

288 82.00000 202405

289 83.00000 111348

290 84.00000 151682

291 85.00000 99919

292 86.00000 46376

293 87.00000 40184

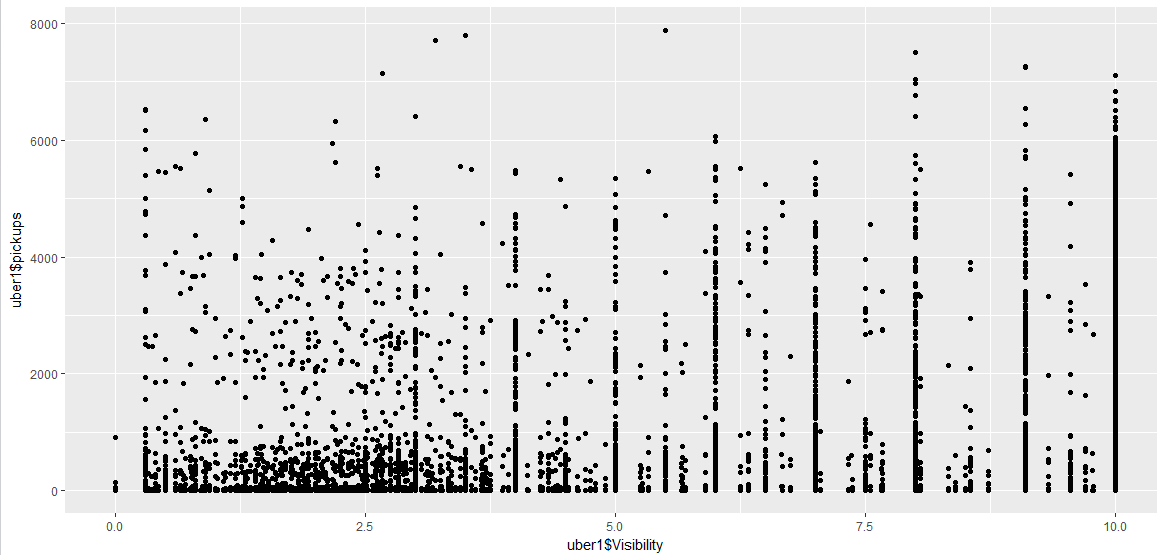
294 88.00000 49481

295 89.00000 28540

#plotting Visibility VS number of pickups

vis.pick <- ggplot(data = uber1,aes(x=uber1$Visibility,y=uber1$pickups))

vis.pick+geom\_jitter()



> aggregate(uber1$pickups,by=list(uber1$Visibility),sum)

Group.1 x

1 0.0000000 1115

2 0.3000000 91438

3 0.3333333 3485

4 0.3666667 2983

5 0.4000000 7035

6 0.4333333 6970

7 0.5000000 23344

8 0.6000000 13598

9 0.6500000 11587

10 0.6666667 4783

11 0.6800000 2308

12 0.7000000 917

13 0.7500000 7597

14 0.7666667 9375

15 0.8000000 22882

16 0.8500000 0

17 0.8666667 5693

18 0.8800000 4918

19 0.9000000 16771

20 0.9400000 12019

21 1.0000000 7488

22 1.0250000 2573

23 1.0800000 2406

24 1.1000000 3221

25 1.1500000 7191

26 1.2000000 12894

27 1.2200000 426

28 1.2500000 0

29 1.2666667 20700

30 1.3000000 9935

31 1.3250000 2941

32 1.3500000 3724

33 1.4000000 11139

34 1.4200000 4426

35 1.4333333 3264

36 1.4500000 9865

37 1.4600000 5307

38 1.4750000 3145

39 1.4833333 3316

40 1.5000000 2889

41 1.5250000 4008

42 1.5333333 712

43 1.5750000 8844

44 1.6000000 4552

45 1.6200000 4607

46 1.6500000 15881

47 1.6666667 7897

48 1.7000000 9046

49 1.7500000 7696

50 1.7600000 4387

51 1.7666667 10533

52 1.8000000 3800

53 1.8250000 11094

54 1.8600000 4208

55 1.8666667 8240

56 1.8750000 3104

57 1.8800000 2347

58 1.9000000 3688

59 1.9200000 2077

60 1.9333333 25099

61 1.9500000 0

62 1.9600000 3694

63 2.0000000 19933

64 2.0250000 6072

65 2.0333333 3278

66 2.0600000 4903

67 2.0750000 2715

68 2.0833333 4464

69 2.1000000 4102

70 2.1200000 9579

71 2.1250000 897

72 2.1666667 8382

73 2.1833333 1950

74 2.2000000 18467

75 2.2200000 4804

76 2.2500000 20775

77 2.2600000 3078

78 2.2666667 11691

79 2.3000000 5061

80 2.3333333 10222

81 2.3600000 2550

82 2.3750000 13047

83 2.4000000 4987

84 2.4200000 3259

85 2.4333333 8520

86 2.4600000 2502

87 2.4714286 3297

88 2.4857143 2247

89 2.5000000 35559

90 2.5166667 4716

91 2.5250000 1475

92 2.5750000 6960

93 2.6000000 14860

94 2.6250000 15901

95 2.6428571 10476

96 2.6600000 3493

97 2.6666667 32374

98 2.7000000 5690

99 2.7500000 37693

100 2.8000000 5226

101 2.8333333 44774

102 2.8750000 5706

103 2.9000000 7681

104 2.9166667 1340

105 2.9500000 2748

106 2.9600000 4387

107 3.0000000 139188

108 3.0600000 3552

109 3.1000000 3884

110 3.1250000 9164

111 3.1428571 1609

112 3.1666667 4215

113 3.2000000 10510

114 3.2500000 14508

115 3.2666667 2539

116 3.3333333 6336

117 3.3600000 6795

118 3.4000000 2443

119 3.4500000 8643

120 3.5000000 47368

121 3.5600000 7027

122 3.6000000 1461

123 3.6666667 17842

124 3.7000000 3430

125 3.7166667 642

126 3.7500000 5814

127 3.8750000 5289

128 3.9333333 4554

129 4.0000000 204533

130 4.1250000 3065

131 4.2500000 9585

132 4.2666667 3864

133 4.3333333 14215

134 4.3750000 3978

135 4.4000000 2959

136 4.4200000 4502

137 4.4500000 6539

138 4.5000000 37179

139 4.5333333 3367

140 4.6250000 3935

141 4.7000000 4454

142 4.7500000 2418

143 4.8000000 610

144 4.9000000 1109

145 5.0000000 183136

146 5.2500000 5588

147 5.2666667 542

148 5.3333333 8146

149 5.5000000 52052

150 5.6666667 7594

151 5.7000000 3426

152 5.9000000 10403

153 6.0000000 322163

154 6.2500000 11848

155 6.3333333 28812

156 6.5000000 58555

157 6.6666667 13045

158 6.7500000 3308

159 7.0000000 372135

160 7.0500000 1506

161 7.3333333 2909

162 7.3666667 859

163 7.5000000 37490

164 7.5500000 9978

165 7.6666667 12895

166 8.0000000 444254

167 8.0250000 4388

168 8.0500000 17488

169 8.3333333 2804

170 8.4000000 836

171 8.5000000 2131

172 8.5500000 19226

173 8.7333333 1578

174 9.1000000 555787

175 9.3333333 7438

176 9.5500000 41647

177 9.7000000 10881

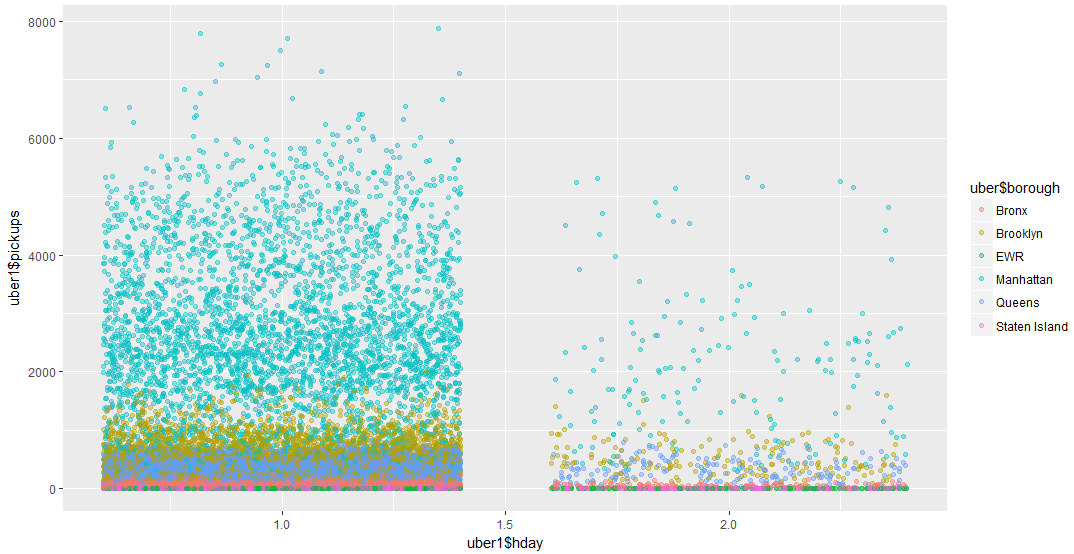
178 9.7750000 3716

179 10.0000000 10454801

#plotting holiday VS number of pickups from various locations

hldy.pick <- ggplot(data = uber1,aes(x=uber1$hday,y=uber1$pickups))

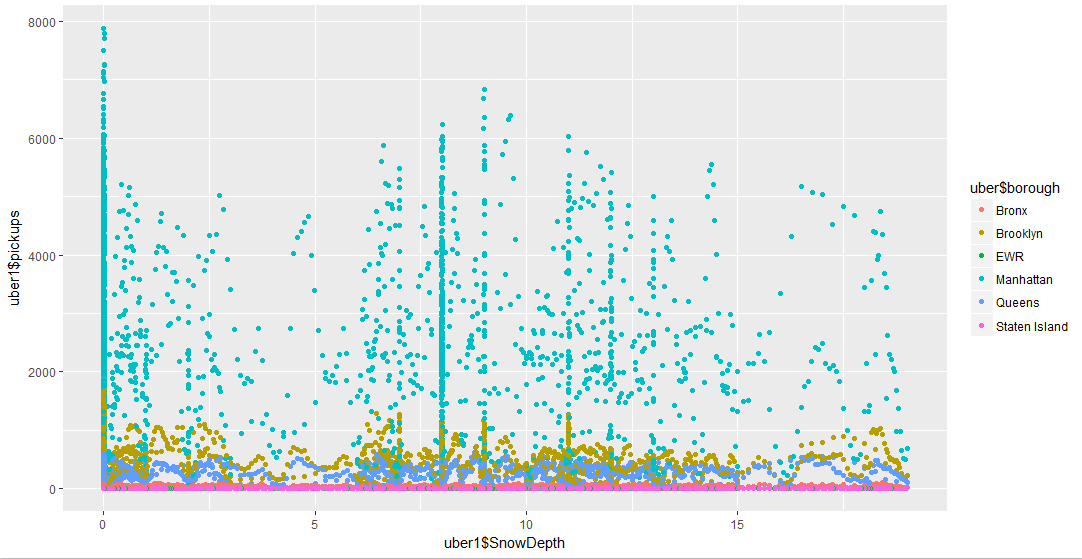
hldy.pick+geom\_jitter(aes(colour=uber$borough),aplha=0.4)



#plotting snowdepth VS number of pickups

snowd.pick <- ggplot(data = uber1,aes(x=uber1$SnowDepth,y=uber1$pickups))

snowd.pick+geom\_jitter(aes(colour=uber$borough))



|  |
| --- |
| > #running T-test for target variable with all the independent variable  > t.test(uber1$pickups,uber1$borough, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber1$borough  t = 84.613, df = 26057, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  531.1269 556.3173  sample estimates:  mean of x mean of y  547.2221 3.5000  > t.test(uber1$pickups,uber$Wind.Speed, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber$Wind.Speed  t = 84.224, df = 26058, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  528.6268 553.8173  sample estimates:  mean of x mean of y  547.222082 6.000039  > t.test(uber1$pickups,uber$Visibility, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber$Visibility  t = 83.785, df = 26057, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  525.8068 550.9973  sample estimates:  mean of x mean of y  547.222082 8.820027  > t.test(uber1$pickups,uber$temp, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber$temp  t = 77.754, df = 26076, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  487.1356 512.3306  sample estimates:  mean of x mean of y  547.2221 47.4890  > t.test(uber1$pickups,uber$DewPoint, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber$DewPoint  t = 80.371, df = 26079, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  503.9698 529.1655  sample estimates:  mean of x mean of y  547.22208 30.65443  > t.test(uber1$pickups,uber$Sea.Press, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber$Sea.Press  t = -73.231, df = 26060, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  -483.1863 -457.9952  sample estimates:  mean of x mean of y  547.2221 1017.8128  > t.test(uber1$pickups,uber$SnowDepth, type='pearson')  Welch Two Sample t-test  data: uber1$pickups and uber$SnowDepth  t = 84.763, df = 26058, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  532.0903 557.2809  sample estimates:  mean of x mean of y  547.222082 2.536496  > t.test(uber1$pickups,as.numeric(uber$hday), type='pearson')  Welch Two Sample t-test  data: uber1$pickups and as.numeric(uber$hday)  t = 84.997, df = 26057, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  533.5884 558.7788  sample estimates:  mean of x mean of y  547.222082 1.038453  > t.test(uber1$pickups,as.numeric(uber$Time), type='pearson')  Welch Two Sample t-test  data: uber1$pickups and as.numeric(uber$Time)  t = 83.211, df = 26059, p-value < 2.2e-16  alternative hypothesis: true difference in means is not equal to 0  95 percent confidence interval:  522.1253 547.3163  sample estimates:  mean of x mean of y  547.22208 12.50127 |
|  |
| |  | | --- | | > | |

|  |
| --- |
| > # random forest on uber dataset  > uber.frst <- read\_excel("E:/Data Analytics Internship/capstone project/random forest/uber\_nyc\_enriched.xlsx")  > #reading the updated excel file withe a new catagorical attribute for number of pickups  > #analysing the data summary  > str(uber.frst)  Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 29101 obs. of 13 variables:  $ pickup\_dt : POSIXct, format: "2015-01-01 00:59:59" "2015-01-01 00:59:59" "2015-01-01 00:59:59" "2015-01-01 00:59:59" ...  $ Date : POSIXct, format: "2015-01-01" "2015-01-01" "2015-01-01" "2015-01-01" ...  $ Time : POSIXct, format: "1899-12-30 01:00:00" "1899-12-30 01:00:00" "1899-12-30 01:00:00" "1899-12-30 01:00:00" ...  $ borough : chr "Bronx" "Brooklyn" "EWR" "Manhattan" ...  $ pickups : num 152 1519 0 5258 405 ...  $ Cat\_Pick : chr "Low" "Average" "Low" "High" ...  $ Wind Speed: num 5 5 5 5 5 5 5 3 3 3 ...  $ Visibility: num 10 10 10 10 10 10 10 10 10 10 ...  $ temp : num 30 30 30 30 30 30 30 30 30 30 ...  $ DewPoint : num 7 7 7 7 7 7 7 6 6 6 ...  $ Sea Press : num 1024 1024 1024 1024 1024 ...  $ SnowDepth : num 0 0 0 0 0 0 0 0 0 0 ...  $ hday : chr "Y" "Y" "Y" "Y" ...  > describe(uber.frst)  vars n mean sd median trimmed mad min max range skew kurtosis se  pickup\_dt\* 1 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  Date\* 2 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  Time\* 3 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  borough\* 4 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  pickups 5 29101 490.22 995.65 54.0 219.75 80.06 0.0 7883.0 7883 2.98 9.27 5.84  Cat\_Pick\* 6 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  Wind Speed 7 29101 5.98 3.70 6.0 5.84 2.97 0.0 21.0 21 0.42 0.42 0.02  Visibility 8 29101 8.82 2.44 10.0 9.47 0.00 0.0 10.0 10 -2.04 2.90 0.01  temp 9 29101 47.67 19.81 46.0 47.54 23.72 2.0 89.0 87 0.06 -1.04 0.12  DewPoint 10 29101 30.82 21.28 30.0 30.89 26.69 -16.0 73.0 89 0.02 -1.04 0.12  Sea Press 11 29101 1017.82 7.77 1018.2 1017.79 7.71 991.4 1043.4 52 0.05 0.07 0.05  SnowDepth 12 29101 2.53 4.52 0.0 1.53 0.00 0.0 19.0 19 1.59 1.31 0.03  hday\* 13 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  There were 15 warnings (use warnings() to see them)  > #remong the number of pickups column to avaoid confusion  > uber.frst <- uber.frst[,c(2:4,6:13)]  > describe(uber.frst)  vars n mean sd median trimmed mad min max range skew kurtosis se  Date\* 1 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  Time\* 2 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  borough\* 3 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  Cat\_Pick\* 4 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  Wind Speed 5 29101 5.98 3.70 6.0 5.84 2.97 0.0 21.0 21 0.42 0.42 0.02  Visibility 6 29101 8.82 2.44 10.0 9.47 0.00 0.0 10.0 10 -2.04 2.90 0.01  temp 7 29101 47.67 19.81 46.0 47.54 23.72 2.0 89.0 87 0.06 -1.04 0.12  DewPoint 8 29101 30.82 21.28 30.0 30.89 26.69 -16.0 73.0 89 0.02 -1.04 0.12  Sea Press 9 29101 1017.82 7.77 1018.2 1017.79 7.71 991.4 1043.4 52 0.05 0.07 0.05  SnowDepth 10 29101 2.53 4.52 0.0 1.53 0.00 0.0 19.0 19 1.59 1.31 0.03  hday\* 11 29101 NaN NA NA NaN NA Inf -Inf -Inf NA NA NA  There were 13 warnings (use warnings() to see them)  > uber.frst$Date <- as.Date(uber.frst$Date)  > uber.frst$Cat\_Pick <- as.factor(uber.frst$Cat\_Pick)  > uber.frst$borough <- as.factor(uber.frst$borough)  > uber.frst$hday <- as.factor(uber.frst$hday)  >  > uber.frst1 <- uber.frst  > uber.frst1$Time <- as.numeric(uber.frst1$Time)  > uber.frst1$Date <- as.numeric(uber.frst1$Date)  > uber.frst1$borough <- as.numeric(uber.frst1$borough)  >  > uber.frst1$hday <- as.numeric(uber.frst1$hday)  > uber.frst1$`Wind Speed` <- as.numeric(uber.frst1$`Wind Speed`)  > str(uber.frst1)  Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 29101 obs. of 11 variables:  $ Date : num 16436 16436 16436 16436 16436 ...  $ Time : num -2.21e+09 -2.21e+09 -2.21e+09 -2.21e+09 -2.21e+09 ...  $ borough : num 1 2 3 4 6 7 5 1 2 3 ...  $ Cat\_Pick : Factor w/ 4 levels "Above Average",..: 4 2 4 3 4 4 4 4 2 4 ...  $ Wind Speed: num 5 5 5 5 5 5 5 3 3 3 ...  $ Visibility: num 10 10 10 10 10 10 10 10 10 10 ...  $ temp : num 30 30 30 30 30 30 30 30 30 30 ...  $ DewPoint : num 7 7 7 7 7 7 7 6 6 6 ...  $ Sea Press : num 1024 1024 1024 1024 1024 ...  $ SnowDepth : num 0 0 0 0 0 0 0 0 0 0 ...  $ hday : num 2 2 2 2 2 2 2 2 2 2 ...  > uber.frst1$borough <- as.factor(uber.frst1$borough)  > uber.frst1$Cat\_Pick <- as.factor(uber.frst1$Cat\_Pick)  > uber.frst1$hday <- as.factor(uber.frst1$hday)  > uber.frst1$Time <- as.factor(uber.frst1$Time)  > #splitting the data as 25% for training and 75% percent for validation  > rndfrst <- sample.split(Y=uber.frst1$Cat\_Pick,SplitRatio = 0.25)  > train <- uber.frst1[rndfrst,]  > test <- uber.frst1[!rndfrst,]  > str(train)  Classes ‘tbl\_df’, ‘tbl’ and 'data.frame': 7275 obs. of 11 variables:  $ Date : num 16436 16436 16436 16436 16436 ...  $ Time : Factor w/ 24 levels "-2209161600",..: 3 4 5 6 6 8 8 10 11 11 ...  $ borough : Factor w/ 7 levels "1","2","3","4",..: 6 6 3 1 6 1 7 2 1 6 ...  $ Cat\_Pick : Factor w/ 4 levels "Above Average",..: 4 4 4 4 4 4 4 4 4 4 ...  $ Wind Speed: num 3 5 5 5 5 9 9 3 3 3 ...  $ Visibility: num 10 10 10 10 10 10 10 10 10 10 ...  $ temp : num 30 30 29 28 28 28 28 27 27 27 ...  $ DewPoint : num 6 8 9 9 9 10 10 7 6 6 ...  $ Sea Press : num 1023 1022 1022 1022 1022 ...  $ SnowDepth : num 0 0 0 0 0 0 0 0 0 0 ...  $ hday : Factor w/ 2 levels "1","2": 2 2 2 2 2 2 2 2 2 2 ...  > train <- train[,2:13]#removing the dates column  Error: Invalid column indexes: 12, 13  > describe(train)  vars n mean sd median trimmed mad min max range skew kurtosis se  Date 1 7275 16526.68 52.20 16527.0 16526.82 66.72 16436.0 16616.00 180.00 -0.01 -1.21 0.61  Time\* 2 7275 12.65 6.93 13.0 12.68 8.90 1.0 24.00 23.00 -0.04 -1.20 0.08  borough\* 3 7275 3.96 2.03 4.0 3.95 2.97 1.0 7.00 6.00 0.06 -1.29 0.02  Cat\_Pick\* 4 7275 3.50 0.96 4.0 3.69 0.00 1.0 4.00 3.00 -1.52 0.67 0.01  Wind Speed 5 7275 5.96 3.69 6.0 5.81 2.97 0.0 21.00 21.00 0.43 0.47 0.04  Visibility 6 7275 8.81 2.45 10.0 9.47 0.00 0.0 10.00 10.00 -2.03 2.83 0.03  temp 7 7275 47.86 20.04 46.0 47.82 25.20 2.0 89.00 87.00 0.03 -1.05 0.23  DewPoint 8 7275 30.98 21.49 30.0 31.07 26.69 -16.0 73.00 89.00 0.01 -1.03 0.25  Sea Press 9 7275 1017.84 7.69 1018.2 1017.78 7.71 991.4 1043.40 52.00 0.08 0.07 0.09  SnowDepth 10 7275 2.60 4.57 0.0 1.61 0.00 0.0 18.96 18.96 1.53 1.04 0.05  hday\* 11 7275 1.04 0.19 1.0 1.00 0.00 1.0 2.00 1.00 4.78 20.83 0.00  > colnames(train)  [1] "Date" "Time" "borough" "Cat\_Pick" "Wind Speed" "Visibility" "temp" "DewPoint" "Sea Press" "SnowDepth"  [11] "hday"  > colnames(train)[colnames(train)=="Wind Speed"] <- c("wspeed")#updating column names  > colnames(train)[colnames(train)=="Sea Press"] <- c("seapress")#updating column names  > #implementing random forest model  > rfrst.model <- randomForest(train$Cat\_Pick~.,data = train)  > importance(rfrst.model)#checking the importance of variables  MeanDecreaseGini  Date 171.117792  Time 381.076837  borough 1426.069947  wspeed 104.056277  Visibility 54.356105  temp 162.940543  DewPoint 151.602282  seapress 161.416351  SnowDepth 50.348127  hday 7.793066  > varImpPlot(rfrst.model)#ploting importance of variables  >  > test <- test[,2:13]  Error: Invalid column indexes: 12, 13  > colnames(test)[colnames(test)=="Wind Speed"] <- c("wspeed")  > colnames(test)[colnames(test)=="Sea Press"] <- c("seapress")  > #predicting the target variable in the test dataset  > prdictclass <- predict(rfrst.model, test,type = 'class')  > #confusion matrix for predicted vs target variable  > a <- table(prediction=prdictclass,actual=test$Cat\_Pick)  > a  actual  prediction Above Average Average High Low  Above Average 875 198 139 1  Average 350 2682 15 558  High 3 2 10 0  Low 2 666 0 16325  > #accuracy of the model  > accuracy <- sum(diag(a))/sum(a)  > accuracy  [1] 0.9113901 |
|  |
| |  | | --- | | > | |