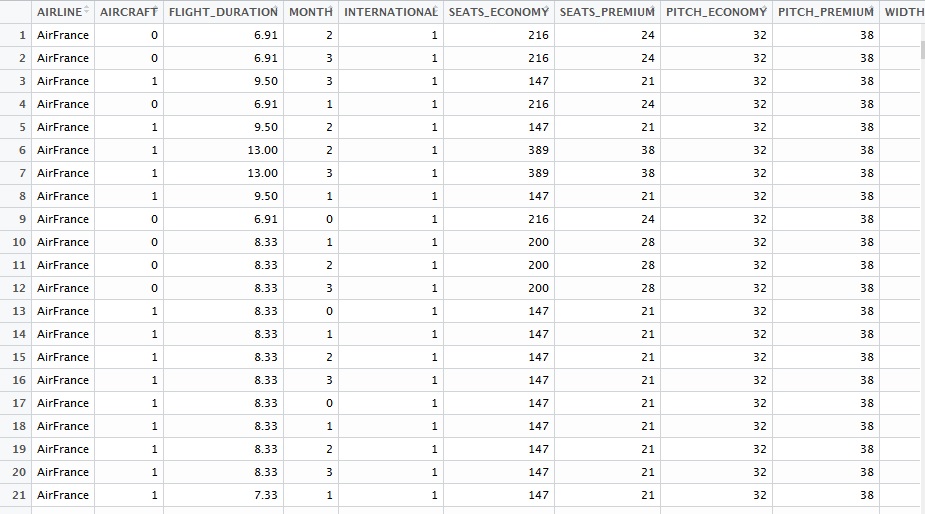
*CODE :*

*data.df <-read.csv(paste("C:/Users/USER/Downloads/SixAirlines.csv", sep=""))#Loading the Data set*

*attach(data.df)#Attaching the Data ste*

*View(data.df)#General view of the entire Data frme*

*OUTPUT :*

**

*CODE :*

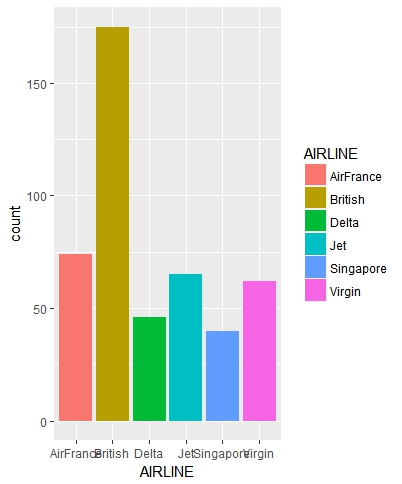
*library(ggplot2)*

*## Loading required package: ggplot2*

*#Seggregating different flights*

*ggplot(data.df, aes(x = AIRLINE, fill = AIRLINE)) + geom\_bar()*

*OUTPUT :*

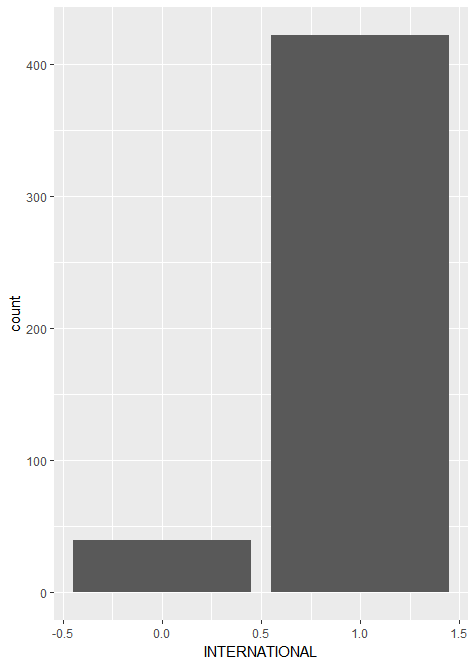
**

*CODE :*

*#Seggregating international and domestic flights*

*ggplot(data.df, aes(x = INTERNATIONAL))+ geom\_bar()*

*OUTPUT :*

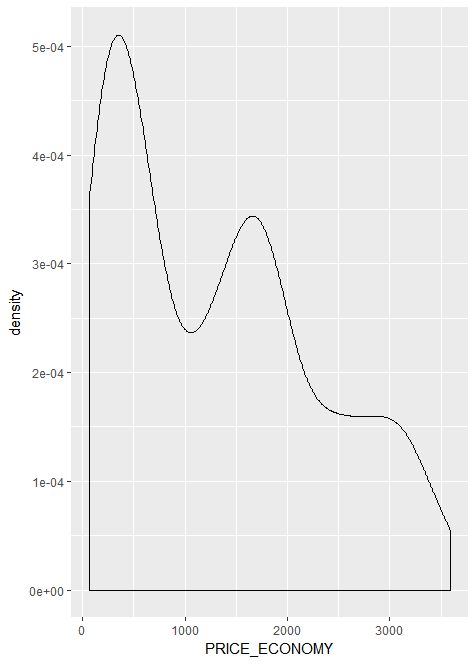
**

*CODE :*

*#Prices of Economy and Premium tickets*

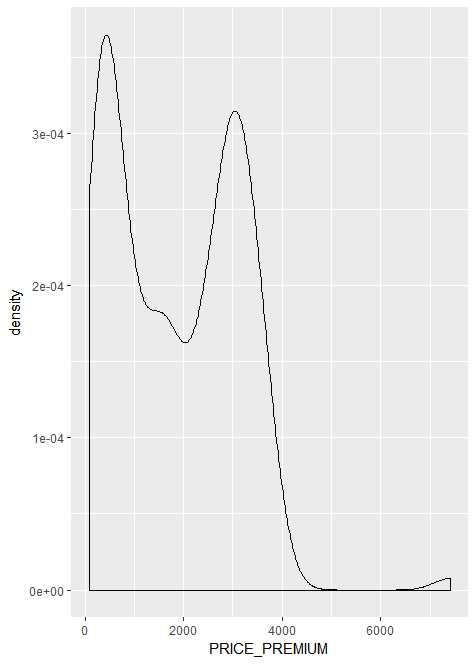
*ggplot(data.df, aes(x = PRICE\_ECONOMY)) + geom\_density()*

*OUTPUT :*

**

*CODE : ggplot(data.df, aes(x = PRICE\_PREMIUM)) + geom\_density()*

*OUTPUT :*

**

*CODE :*

*# A Scatterplot of price economy vs flight hours of travel*

*# ==========*

*plot(FLIGHT\_DURATION,PRICE\_ECONOMY,*

*col="blue",*

*main="Price economy vs flight hours",*

*xlab="Hours", ylab="Price")*

*# Add the sample means to the Scatterplot*

*# ==========*

*abline(h=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")*

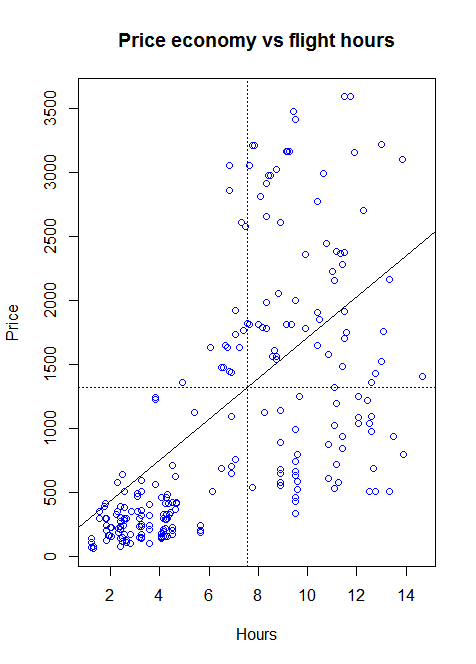
*abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")*

*# Add a regression line*

*# ==========*

*abline(lm(PRICE\_ECONOMY ~ FLIGHT\_DURATION))*

*OUTPUT :*

**

*CODE :*

*# A Scatterplot of price premium vs flight hours of travel*

*# ==========*

*plot(FLIGHT\_DURATION,PRICE\_PREMIUM,*

*col="blue",*

*main="Price economy vs flight hours",*

*xlab="Hours", ylab="Price")*

*# Add the sample means to the Scatterplot*

*# ==========*

*abline(h=mean(PRICE\_PREMIUM), col="dark blue", lty="dotted")*

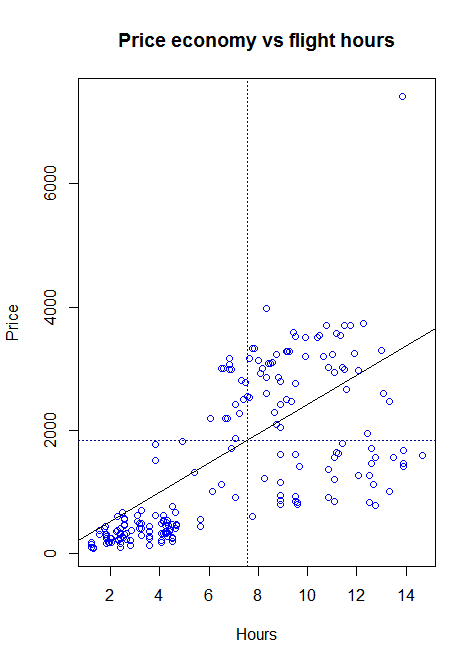
*abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")*

*# Add a regression line*

*# ==========*

*abline(lm(PRICE\_PREMIUM ~ FLIGHT\_DURATION))*

*OUTPUT :*

**

*CODE :*

*#Correlation and Correlation Matrix for Price Economy*

*library(corrplot)*

*library(gplots) # for color interpolation*

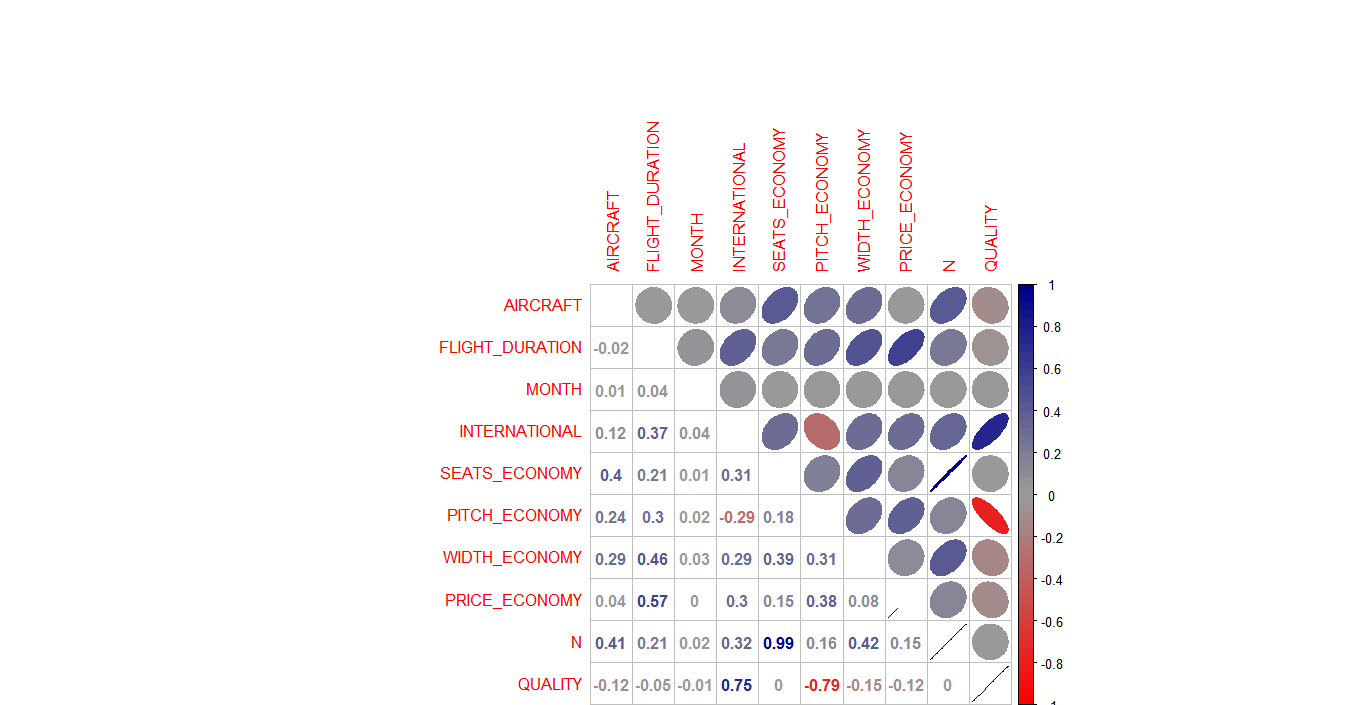
*par(mfrow=c(1, 1))*

*corrplot.mixed(corr=cor(data.df[ , c(2:6, 8,10,12,15,17)], use="complete.obs"),*

*upper="ellipse", tl.pos="lt",*

*col = colorpanel(50, "red", "gray60", "blue4"))*

*OUTPUT :*

**

*CODE :*

*#Correlation and Correlation Matrix for Price Premium*

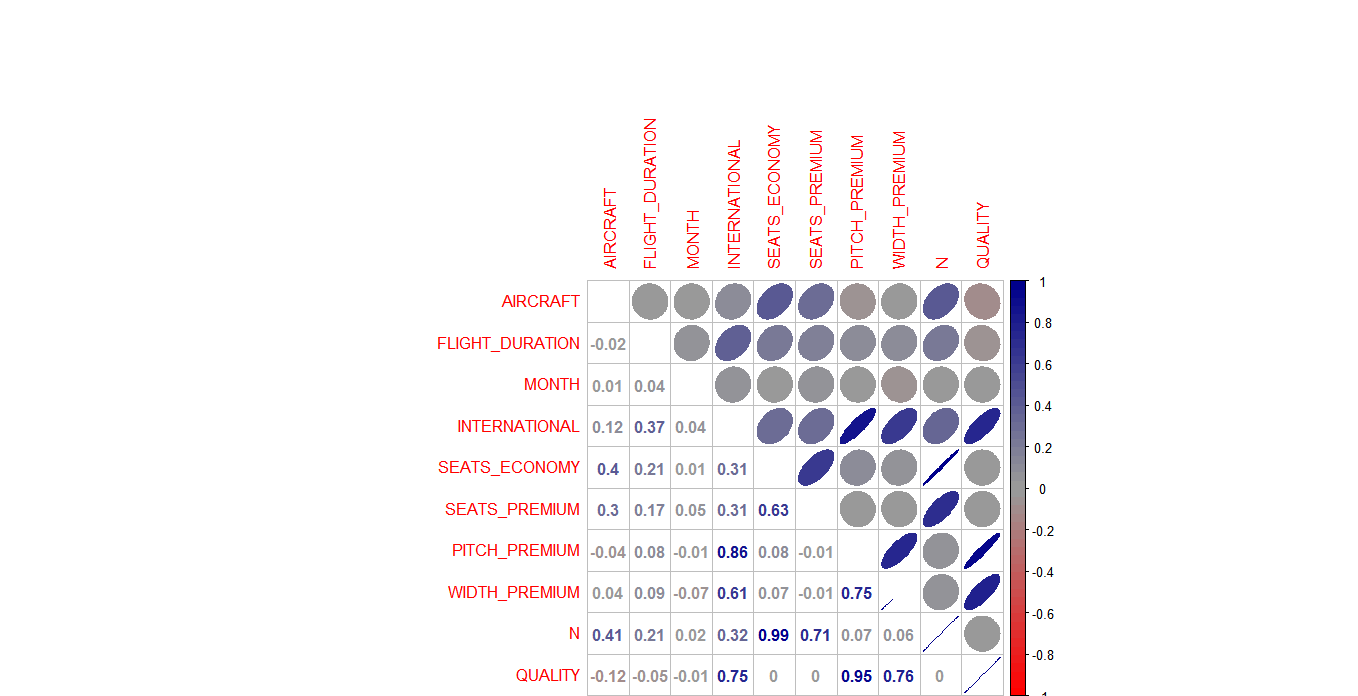
*par(mfrow=c(1, 1))*

*corrplot.mixed(corr=cor(data.df[ , c(2:6, 7,9,11,15,17)], use="complete.obs"),*

*upper="ellipse", tl.pos="lt",*

*col = colorpanel(50, "red", "gray60", "blue4"))*

*OUTPUT :*

**

*CODE :*

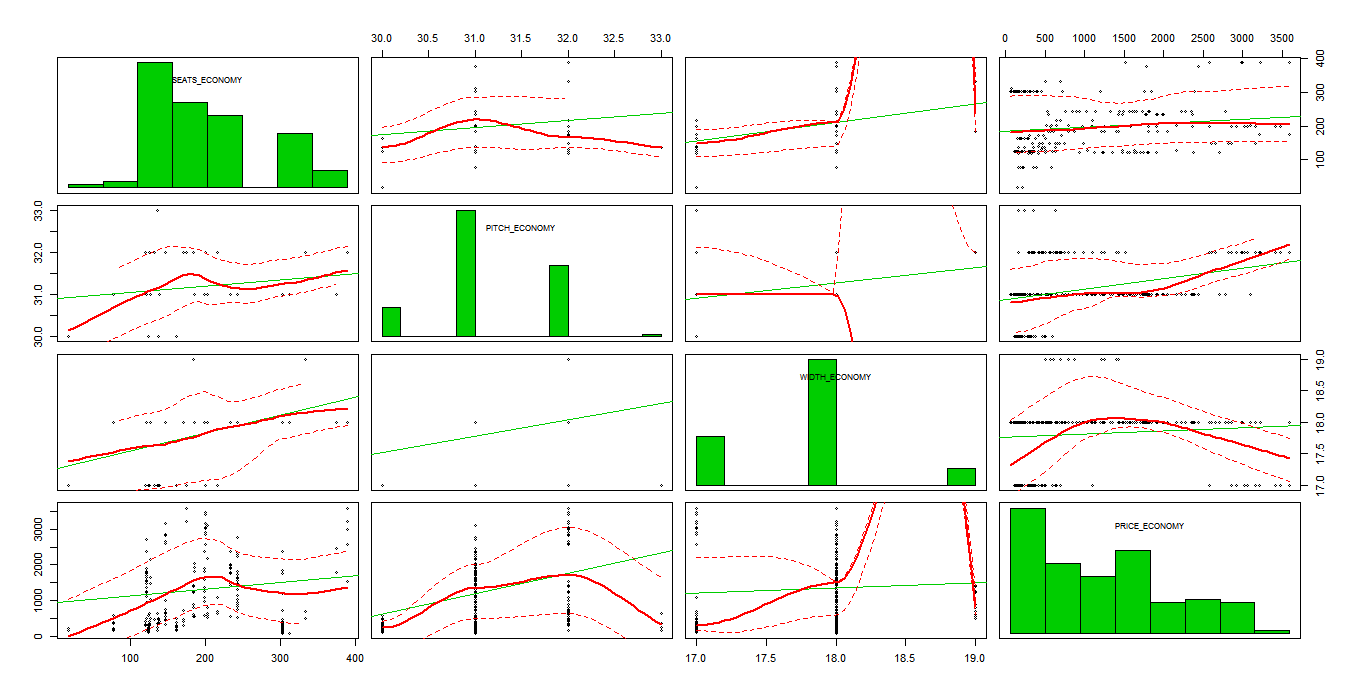
*#Scatter Plot Matrix for Price Economy and Price Premium*

*library(car)*

*scatterplotMatrix(formula = ~ SEATS\_ECONOMY + PITCH\_ECONOMY + WIDTH\_ECONOMY + PRICE\_ECONOMY, cex=0.6,*

*data=data.df, diagonal="histogram")*

*OUTPUT :*

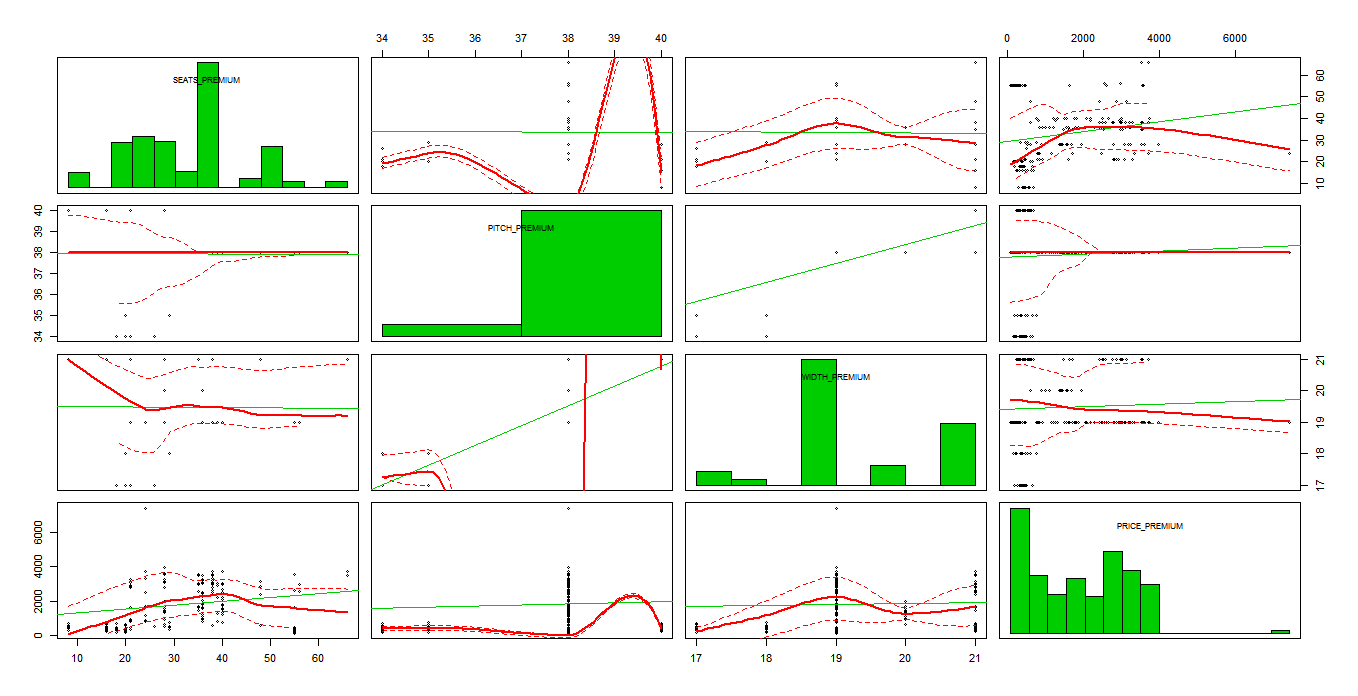
**

*CODE :*

*scatterplotMatrix(formula = ~ SEATS\_PREMIUM + PITCH\_PREMIUM + WIDTH\_PREMIUM + PRICE\_PREMIUM, cex=0.6,*

*data=data.df, diagonal="histogram")*

*OUTPUT :*

**

*CODE :*

*#Calculating correlations between Prices of Economy and Premium in correlation to other factors*

*cor.test(PRICE\_ECONOMY, PITCH\_ECONOMY)*

*cor.test(PRICE\_ECONOMY, WIDTH\_ECONOMY)*

*OUTPUT :*

> cor.test(PRICE\_ECONOMY, PITCH\_ECONOMY)

Pearson's product-moment correlation

data: PRICE\_ECONOMY and PITCH\_ECONOMY

t = 8.8003, df = 460, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.2987210 0.4550742

sample estimates:

cor

0.379605

> cor.test(PRICE\_ECONOMY, WIDTH\_ECONOMY)

Pearson's product-moment correlation

data: PRICE\_ECONOMY and WIDTH\_ECONOMY

t = 1.764, df = 460, p-value = 0.0784

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.009330795 0.171911298

sample estimates:

cor

0.0819679

*CODE :*

*cor.test(PRICE\_PREMIUM, PITCH\_PREMIUM)*

*cor.test(PRICE\_PREMIUM, WIDTH\_PREMIUM)*

*OUTPUT :*

[185] "gray32" "gray33" "gray34" "gray35"

[189] "gray36" "gray37" "gray38" "gray39"

[193] "gray40" "gray41" "gray42" "gray43"

[197] "gray44" "gray45" "gray46" "gray47"

[201] "gray48" "gray49" "gray50" "gray51"

[205] "gray52" "gray53" "gray54" "gray55"

[209] "gray56" "gray57" "gray58" "gray59"

[213] "gray60" "gray61" "gray62" "gray63"

[217] "gray64" "gray65" "gray66" "gray67"

[221] "gray68" "gray69" "gray70" "gray71"

[225] "gray72" "gray73" "gray74" "gray75"

[229] "gray76" "gray77" "gray78" "gray79"

[233] "gray80" "gray81" "gray82" "gray83"

[237] "gray84" "gray85" "gray86" "gray87"

[241] "gray88" "gray89" "gray90" "gray91"

[245] "gray92" "gray93" "gray94" "gray95"

[249] "gray96" "gray97" "gray98" "gray99"

[253] "gray100" "green" "green1" "green2"

[257] "green3" "green4" "greenyellow" "grey"

[261] "grey0" "grey1" "grey2" "grey3"

[265] "grey4" "grey5" "grey6" "grey7"

[269] "grey8" "grey9" "grey10" "grey11"

[273] "grey12" "grey13" "grey14" "grey15"

[277] "grey16" "grey17" "grey18" "grey19"

[281] "grey20" "grey21" "grey22" "grey23"

[285] "grey24" "grey25" "grey26" "grey27"

[289] "grey28" "grey29" "grey30" "grey31"

[293] "grey32" "grey33" "grey34" "grey35"

[297] "grey36" "grey37" "grey38" "grey39"

[301] "grey40" "grey41" "grey42" "grey43"

[305] "grey44" "grey45" "grey46" "grey47"

[309] "grey48" "grey49" "grey50" "grey51"

[313] "grey52" "grey53" "grey54" "grey55"

[317] "grey56" "grey57" "grey58" "grey59"

[321] "grey60" "grey61" "grey62" "grey63"

[325] "grey64" "grey65" "grey66" "grey67"

[329] "grey68" "grey69" "grey70" "grey71"

[333] "grey72" "grey73" "grey74" "grey75"

[337] "grey76" "grey77" "grey78" "grey79"

[341] "grey80" "grey81" "grey82" "grey83"

[345] "grey84" "grey85" "grey86" "grey87"

[349] "grey88" "grey89" "grey90" "grey91"

[353] "grey92" "grey93" "grey94" "grey95"

[357] "grey96" "grey97" "grey98" "grey99"

[361] "grey100" "honeydew" "honeydew1" "honeydew2"

[365] "honeydew3" "honeydew4" "hotpink" "hotpink1"

[369] "hotpink2" "hotpink3" "hotpink4" "indianred"

[373] "indianred1" "indianred2" "indianred3" "indianred4"

[377] "ivory" "ivory1" "ivory2" "ivory3"

[381] "ivory4" "khaki" "khaki1" "khaki2"

[385] "khaki3" "khaki4" "lavender" "lavenderblush"

[389] "lavenderblush1" "lavenderblush2" "lavenderblush3" "lavenderblush4"

[393] "lawngreen" "lemonchiffon" "lemonchiffon1" "lemonchiffon2"

[397] "lemonchiffon3" "lemonchiffon4" "lightblue" "lightblue1"

[401] "lightblue2" "lightblue3" "lightblue4" "lightcoral"

[405] "lightcyan" "lightcyan1" "lightcyan2" "lightcyan3"

[409] "lightcyan4" "lightgoldenrod" "lightgoldenrod1" "lightgoldenrod2"

[413] "lightgoldenrod3" "lightgoldenrod4" "lightgoldenrodyellow" "lightgray"

[417] "lightgreen" "lightgrey" "lightpink" "lightpink1"

[421] "lightpink2" "lightpink3" "lightpink4" "lightsalmon"

[425] "lightsalmon1" "lightsalmon2" "lightsalmon3" "lightsalmon4"

[429] "lightseagreen" "lightskyblue" "lightskyblue1" "lightskyblue2"

[433] "lightskyblue3" "lightskyblue4" "lightslateblue" "lightslategray"

[437] "lightslategrey" "lightsteelblue" "lightsteelblue1" "lightsteelblue2"

[441] "lightsteelblue3" "lightsteelblue4" "lightyellow" "lightyellow1"

[445] "lightyellow2" "lightyellow3" "lightyellow4" "limegreen"

[449] "linen" "magenta" "magenta1" "magenta2"

[453] "magenta3" "magenta4" "maroon" "maroon1"

[457] "maroon2" "maroon3" "maroon4" "mediumaquamarine"

[461] "mediumblue" "mediumorchid" "mediumorchid1" "mediumorchid2"

[465] "mediumorchid3" "mediumorchid4" "mediumpurple" "mediumpurple1"

[469] "mediumpurple2" "mediumpurple3" "mediumpurple4" "mediumseagreen"

[473] "mediumslateblue" "mediumspringgreen" "mediumturquoise" "mediumvioletred"

[477] "midnightblue" "mintcream" "mistyrose" "mistyrose1"

[481] "mistyrose2" "mistyrose3" "mistyrose4" "moccasin"

[485] "navajowhite" "navajowhite1" "navajowhite2" "navajowhite3"

[489] "navajowhite4" "navy" "navyblue" "oldlace"

[493] "olivedrab" "olivedrab1" "olivedrab2" "olivedrab3"

[497] "olivedrab4" "orange" "orange1" "orange2"

[501] "orange3" "orange4" "orangered" "orangered1"

[505] "orangered2" "orangered3" "orangered4" "orchid"

[509] "orchid1" "orchid2" "orchid3" "orchid4"

[513] "palegoldenrod" "palegreen" "palegreen1" "palegreen2"

[517] "palegreen3" "palegreen4" "paleturquoise" "paleturquoise1"

[521] "paleturquoise2" "paleturquoise3" "paleturquoise4" "palevioletred"

[525] "palevioletred1" "palevioletred2" "palevioletred3" "palevioletred4"

[529] "papayawhip" "peachpuff" "peachpuff1" "peachpuff2"

[533] "peachpuff3" "peachpuff4" "peru" "pink"

[537] "pink1" "pink2" "pink3" "pink4"

[541] "plum" "plum1" "plum2" "plum3"

[545] "plum4" "powderblue" "purple" "purple1"

[549] "purple2" "purple3" "purple4" "red"

[553] "red1" "red2" "red3" "red4"

[557] "rosybrown" "rosybrown1" "rosybrown2" "rosybrown3"

[561] "rosybrown4" "royalblue" "royalblue1" "royalblue2"

[565] "royalblue3" "royalblue4" "saddlebrown" "salmon"

[569] "salmon1" "salmon2" "salmon3" "salmon4"

[573] "sandybrown" "seagreen" "seagreen1" "seagreen2"

[577] "seagreen3" "seagreen4" "seashell" "seashell1"

[581] "seashell2" "seashell3" "seashell4" "sienna"

[585] "sienna1" "sienna2" "sienna3" "sienna4"

[589] "skyblue" "skyblue1" "skyblue2" "skyblue3"

[593] "skyblue4" "slateblue" "slateblue1" "slateblue2"

[597] "slateblue3" "slateblue4" "slategray" "slategray1"

[601] "slategray2" "slategray3" "slategray4" "slategrey"

[605] "snow" "snow1" "snow2" "snow3"

[609] "snow4" "springgreen" "springgreen1" "springgreen2"

[613] "springgreen3" "springgreen4" "steelblue" "steelblue1"

[617] "steelblue2" "steelblue3" "steelblue4" "tan"

[621] "tan1" "tan2" "tan3" "tan4"

[625] "thistle" "thistle1" "thistle2" "thistle3"

[629] "thistle4" "tomato" "tomato1" "tomato2"

[633] "tomato3" "tomato4" "turquoise" "turquoise1"

[637] "turquoise2" "turquoise3" "turquoise4" "violet"

[641] "violetred" "violetred1" "violetred2" "violetred3"

[645] "violetred4" "wheat" "wheat1" "wheat2"

[649] "wheat3" "wheat4" "whitesmoke" "yellow"

[653] "yellow1" "yellow2" "yellow3" "yellow4"

[657] "yellowgreen"

> hist(AIRCRAFT,

+ main="Airline origin",

+ xlab="Countries",

+ ylab="Origin",

+ xlim=c(50,200), ylim=c(0,500), # add limits to the axes

+ breaks=10, # more columns

+ col="lightblue") # color the bars

> hist(AIRCRAFT,

+ main="Product 1 Weekly Sales Frequencies, All Stores",

+ xlab="Product 1 Sales", ylab="Relative frequency",

+ breaks=30, col="lightblue", freq=FALSE)

> lines(density(store.df$p1sales, bw=10), # bw = smoothing

+ type="l", col="darkred", lwd=2) # lwd = line width

Error in density(store.df$p1sales, bw = 10) : object 'store.df' not found

> lines(density(AIRCRAFT, bw=10), # bw = smoothing

+ type="l", col="darkred", lwd=2) # lwd = line width

> hist(PRICE\_ECONOMY,

+ main="Product 1 Weekly Sales Frequencies, All Stores",

+ xlab="Product 1 Sales", ylab="Relative frequency",

+ breaks=30, col="lightblue", freq=FALSE)

> lines(density(PRICE\_ECONOMY, bw=10), # bw = smoothing

+ type="l", col="darkred", lwd=2) # lwd = line width

> require(ggplot2)

Loading required package: ggplot2

Attaching package: ‘ggplot2’

The following objects are masked from ‘package:psych’:

%+%, alpha

> ggplot(data.df, aes(x = AIRCRAFT)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = AIRLINE)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = AIRCRAFT)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = AIRCRAFT)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = INTERNATIONAL)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = QUALITY)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = MONTH)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = MONTH)) + geom\_bar()

> ggplot(data.df, aes(x = AIRLINE, fill = MONTH)) + geom\_bar(position = dodge)

Error in layer(data = data, mapping = mapping, stat = stat, geom = GeomBar, :

object 'dodge' not found

> ggplot(data.df, aes(x = AIRLINE, fill = MONTH)) + geom\_bar(position = "dodge")

> ggplot(data.df, aes(x = AIRLINE, fill = AIRCRAFT)) + geom\_bar(position = "dodge")

> ggplot(data.df, aes(x = AIRLINE, fill = AIRLINE)) + geom\_bar()

> ggplot(data.df, aes(x = INTERNATIONAL))+ geom\_bar()

> ggplot(data.df, aes(x = PRICE\_ECONOMY)) + geom\_density()

> ggplot(data.df, aes(x = PRICE\_PREMIUM)) + geom\_density()

> library(psych)

> describe(data.df)

vars n mean sd median trimmed mad min max range skew kurtosis se

AIRLINE\* 1 462 3.02 1.65 2.00 2.90 1.48 1.00 6.00 5.00 0.59 -0.95 0.08

AIRCRAFT 2 462 0.33 0.47 0.00 0.28 0.00 0.00 1.00 1.00 0.74 -1.46 0.02

FLIGHT\_DURATION 3 462 7.55 3.54 7.75 7.54 4.82 1.25 14.66 13.41 -0.05 -1.12 0.16

MONTH 4 462 1.67 1.05 2.00 1.71 1.48 0.00 3.00 3.00 -0.16 -1.20 0.05

INTERNATIONAL 5 462 0.91 0.28 1.00 1.00 0.00 0.00 1.00 1.00 -2.93 6.60 0.01

SEATS\_ECONOMY 6 462 200.71 77.96 185.00 193.76 85.99 17.00 389.00 372.00 0.61 -0.26 3.63

SEATS\_PREMIUM 7 462 33.54 13.26 36.00 33.20 11.86 8.00 66.00 58.00 0.25 -0.46 0.62

PITCH\_ECONOMY 8 462 31.21 0.66 31.00 31.25 0.00 30.00 33.00 3.00 -0.03 -0.38 0.03

PITCH\_PREMIUM 9 462 37.92 1.32 38.00 38.06 0.00 34.00 40.00 6.00 -1.48 3.43 0.06

WIDTH\_ECONOMY 10 462 17.83 0.56 18.00 17.81 0.00 17.00 19.00 2.00 -0.03 -0.12 0.03

WIDTH\_PREMIUM 11 462 19.48 1.10 19.00 19.54 0.00 17.00 21.00 4.00 -0.09 -0.34 0.05

PRICE\_ECONOMY 12 462 1317.06 989.81 1224.00 1231.30 1163.84 65.00 3593.00 3528.00 0.52 -0.88 46.05

PRICE\_PREMIUM 13 462 1832.35 1289.97 1710.00 1782.94 1852.51 86.00 7414.00 7328.00 0.51 0.41 60.01

PRICE\_RELATIVE 14 462 0.49 0.45 0.38 0.43 0.42 0.02 1.89 1.87 1.14 0.61 0.02

N 15 462 234.25 86.88 227.00 227.69 90.44 38.00 441.00 403.00 0.61 -0.44 4.04

LAMBDA 16 462 0.15 0.06 0.13 0.14 0.03 0.05 0.55 0.50 2.70 14.02 0.00

QUALITY 17 462 6.72 1.78 7.00 6.79 0.00 2.00 10.00 8.00 -0.51 1.67 0.08

> ggplot(data.df, aes(x = PRICE\_ECONOMY, color = PRICE\_PREMIUM)) + geom\_density()

> ggplot(data.df, aes(x = PRICE\_ECONOMY, color = PRICE\_PREMIUM)) + geom\_density()

> ggplot(data.df, aes(x = PRICE\_ECONOMY, color = INTERNATIONAL)) + geom\_density()

> ggplot(data.df, aes(x = PRICE\_ECONOMY, color = INTERNATIONAL)) + geom\_density()

> ggplot(data.df, aes(x = PRICE\_ECONOMY, color = AIRCRAFT)) + geom\_density()

> plot(PRICE\_ECONOMY,FLIGHT\_DURATION

+ col="blue",

Error: unexpected symbol in:

"plot(PRICE\_ECONOMY,FLIGHT\_DURATION

col"

> xlim=c(15, 55), ylim=c(500, 900),

Error: unexpected ',' in " xlim=c(15, 55),"

> main="Price economy vs flight hours",

Error: unexpected ',' in " main="Price economy vs flight hours","

> xlab="Price", ylab="Hours")

Error: unexpected ',' in " xlab="Price","

> plot(data.df$PRICE\_ECONOMY,data.df$FLIGHT\_DURATION

+ col="blue",

Error: unexpected symbol in:

"plot(data.df$PRICE\_ECONOMY,data.df$FLIGHT\_DURATION

col"

> xlim=c(15, 55), ylim=c(500, 900),

Error: unexpected ',' in " xlim=c(15, 55),"

> main="Price economy vs flight hours",

Error: unexpected ',' in " main="Price economy vs flight hours","

> xlab="Price", ylab="Hours")

Error: unexpected ',' in " xlab="Price","

> plot(data.df$PRICE\_ECONOMY,data.df$FLIGHT\_DURATION,

+ col="blue",

+ xlim=c(15, 55), ylim=c(500, 900),

+ main="Price economy vs flight hours",

+ xlab="Price", ylab="Hours")

> plot(PRICE\_ECONOMY,FLIGHT\_DURATION,

+ col="blue",

+ xlim=c(15, 55), ylim=c(500, 900),

+ main="Price economy vs flight hours",

+ xlab="Price", ylab="Hours")

> plot(FLIGHT\_DURATION,PRICE\_ECONOMY,

+ col="blue",

+ xlim=c(15, 55), ylim=c(500, 900),

+ main="Price economy vs flight hours",

+ xlab="Price", ylab="Hours")

> plot(FLIGHT\_DURATION,PRICE\_ECONOMY,

+ col="blue",

+ main="Price economy vs flight hours",

+ xlab="Price", ylab="Hours")

> abline(v=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")

> abline(h=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

> abline(v=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")

> plot(FLIGHT\_DURATION,PRICE\_ECONOMY,

+ col="blue",

+ main="Price economy vs flight hours",

+ xlab="Hours", ylab="Price")

>

> #5c. Add the sample means to the Scatterplot

> # ==========

> abline(h=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

> abline(v=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")

> abline(lm(PRICE\_ECONOMY ~ FLIGHT\_DURATION))

> abline(h=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")

> abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

> abline(lm(PRICE\_ECONOMY ~ FLIGHT\_DURATION))

> #5b. A better Scatterplot of price economy vs flight hours of travel

> # ==========

> plot(FLIGHT\_DURATION,PRICE\_ECONOMY,

+ col="blue",

+ main="Price economy vs flight hours",

+ xlab="Hours", ylab="Price")

>

> #5c. Add the sample means to the Scatterplot

> # ==========

> abline(h=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")

> abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

>

> #5d. Add a regression line

> # ==========

> abline(lm(PRICE\_ECONOMY ~ FLIGHT\_DURATION))

> plot(FLIGHT\_DURATION,PRICE\_PREMIUM,

+ col="blue",

+ main="Price economy vs flight hours",

+ xlab="Hours", ylab="Price")

> abline(h=mean(PRICE\_PREMIUM), col="dark blue", lty="dotted")

> abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

> abline(lm(PRICE\_PREMIUM ~ FLIGHT\_DURATION))

> library(gplots) # for color interpolation

Attaching package: ‘gplots’

The following object is masked from ‘package:stats’:

lowess

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

> corrplot.mixed(corr=cor(data.df[ , c(1:6, 8,10,12)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

Error in corrplot.mixed(corr = cor(data.df[, c(1:6, 8, 10, 12)], use = "complete.obs"), :

could not find function "corrplot.mixed"

> library(corrplot)

> library(gplots) # for color interpolation

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

> corrplot.mixed(corr=cor(data.df[ , c(1:6, 8,10,12)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

Error in cor(data.df[, c(1:6, 8, 10, 12)], use = "complete.obs") :

'x' must be numeric

> corrplot.mixed(corr=cor(data.df[ , c(2:6, 8,10,12)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

> corrplot.mixed(corr=cor(data.df[ , c(2:6, 8,10,12,15,17)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

>

>

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

> corrplot.mixed(corr=cor(data.df[ , c(2:6, 7,9,11,15,17)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

>

>

> library(car)

Attaching package: ‘car’

The following object is masked from ‘package:psych’:

logit

> scatterplotMatrix(formula = SEATS\_ECONOMY+PITCH\_ECONOMY+WIDTH\_ECONOMY+PRICE\_ECONOMY, cex=0.6,

+ data=data.df, diagonal="histogram")

Error in rownames(x) : argument "x" is missing, with no default

> scatterplotMatrix(formula = ~ SEATS\_ECONOMY + PITCH\_ECONOMY + WIDTH\_ECONOMY + PRICE\_ECONOMY, cex=0.6,

+ data=data.df, diagonal="histogram")

Warning message:

In smoother(x, y, col = col[2], log.x = FALSE, log.y = FALSE, spread = spread, :

could not fit smooth

> scatterplotMatrix(formula = ~ SEATS\_PREMIUM + PITCH\_PREMIUM + WIDTH\_PREMIUM + PRICE\_PREMIUM, cex=0.6,

+ data=data.df, diagonal="histogram")

Warning message:

In smoother(x, y, col = col[2], log.x = FALSE, log.y = FALSE, spread = spread, :

could not fit smooth

> cor.test(PRICE\_ECONOMY, PITCH\_ECONOMY)

Pearson's product-moment correlation

data: PRICE\_ECONOMY and PITCH\_ECONOMY

t = 8.8003, df = 460, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.2987210 0.4550742

sample estimates:

cor

0.379605

> cor.test(PRICE\_ECONOMY, WIDTH\_ECONOMY)

Pearson's product-moment correlation

data: PRICE\_ECONOMY and WIDTH\_ECONOMY

t = 1.764, df = 460, p-value = 0.0784

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.009330795 0.171911298

sample estimates:

cor

0.0819679

> cor.test(PRICE\_PREMIUM, PITCH\_PREMIUM)

Pearson's product-moment correlation

data: PRICE\_PREMIUM and PITCH\_PREMIUM

t = 1.5338, df = 460, p-value = 0.1258

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.02002801 0.16150915

sample estimates:

cor

0.07133125

> cor.test(PRICE\_PREMIUM, WIDTH\_PREMIUM)

Pearson's product-moment correlation

data: PRICE\_PREMIUM and WIDTH\_PREMIUM

t = 1.0592, df = 460, p-value = 0.2901

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.04209336 0.13992426

sample estimates:

cor

0.04932498

> ggplot(data.df)+ geom\_line(aes(x=PRICE\_ECONOMY, y=PITCH\_ECONOMY, color=”red”))

Error: unexpected input in "ggplot(data.df)+ geom\_line(aes(x=PRICE\_ECONOMY, y=PITCH\_ECONOMY, color=”"

> ggplot(data.df, aes(x=PRICE\_ECONOMY, y=PITCH\_ECONOMY, color=”red”))+ geom\_line()

Error: unexpected input in "ggplot(data.df, aes(x=PRICE\_ECONOMY, y=PITCH\_ECONOMY, color=”"

> ggplot(data.df, aes(x=PRICE\_ECONOMY, y=PITCH\_ECONOMY))+ geom\_line()

> ggplot(data.df, aes(x=PRICE\_ECONOMY, y=WIDTH\_ECONOMY))+ geom\_line()

> ggplot(data.df, aes(x=PRICE\_ECONOMY, y=PRICE\_PREMIUM))+ geom\_line()

> ggplot(data.df, aes(x=PRICE\_ECONOMY, y=PRICE\_RELATIVE))+ geom\_line()

> install.packages("Boruta")

Installing package into ‘C:/Users/USER/Documents/R/win-library/3.4’

(as ‘lib’ is unspecified)

also installing the dependency ‘ranger’

Warning in install.packages :

cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/3.4/PACKAGES.rds': HTTP status was '404 Not Found'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/ranger\_0.8.0.zip'

Content type 'application/zip' length 922300 bytes (900 KB)

downloaded 900 KB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/Boruta\_5.2.0.zip'

Content type 'application/zip' length 51663 bytes (50 KB)

downloaded 50 KB

package ‘ranger’ successfully unpacked and MD5 sums checked

package ‘Boruta’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\USER\AppData\Local\Temp\Rtmp2HmsxM\downloaded\_packages

> library(Boruta)

Loading required package: ranger

Warning messages:

1: package ‘Boruta’ was built under R version 3.4.1

2: package ‘ranger’ was built under R version 3.4.1

> set.seed(1234) # for code reproducibility

> response <- data.df$PRICE\_ECONOMY

> bor.results <- Boruta(data.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- eco.df$PRICE\_ECONOMY

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- eco.df$PRICE\_ECONOMY

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

Error in `[.data.frame`(data, , dependent.variable.name) :

undefined columns selected

In addition: Warning message:

In is.na(y) : is.na() applied to non-(list or vector) of type 'NULL'

> plot(bor.results)

> eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- eco.df$Price

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- data.df$PRICE\_ECONOMY

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- data.df$PRICE\_ECONOMY

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> View(eco.df)

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- data.df$PRICE\_ECONOMY

> bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> pre.df <-read.csv(paste("C:/Users/USER/Downloads/prem.csv", sep=""))

> library(Boruta)

> set.seed(1234) # for code reproducibility

> response <- data.df$PRICE\_PREMIUM

> bor.results <- Boruta(pre.df,response,maxRuns=101,doTrace=0)

> plot(bor.results)

> ratio = sample(1:nrow(data.df), size = 0.25\*nrow(data.df))

> Test = data.df[ratio,] #Test dataset 25% of total

> Training = data.df[-ratio,] #Train dataset 75% of total

> dim(Training)

[1] 347 17

> dim(Test)

[1] 115 17

> linear.mod<- lm(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

> summary(linear.mod)

Call:

lm(formula = PRICE\_ECONOMY ~ PITCH\_ECONOMY + WIDTH\_ECONOMY +

FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

Residuals:

Min 1Q Median 3Q Max

-1503.78 -485.28 -16.78 587.93 1655.51

Coefficients:

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

(Intercept) -8566.89 3343.91 -2.562 0.0108 \*

PITCH\_ECONOMY 574.45 100.45 5.719 2.35e-08 \*\*\*

WIDTH\_ECONOMY -580.68 69.37 -8.370 1.50e-15 \*\*\*

FLIGHT\_DURATION 178.61 11.96 14.934 < 2e-16 \*\*\*

QUALITY 200.08 35.49 5.637 3.63e-08 \*\*\*

PRICE\_RELATIVE -872.63 91.76 -9.510 < 2e-16 \*\*\*

---

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

Residual standard error: 649.2 on 341 degrees of freedom

Multiple R-squared: 0.569, Adjusted R-squared: 0.5626

F-statistic: 90.02 on 5 and 341 DF, p-value: < 2.2e-16

> RMSE <- function(x,y)

+ {

+ a <- sqrt(sum((log(x)-log(y))^2)/length(y))

+ return(a)

+ }

> predict<- predict(linear.mod, Test)

> RMSE0<- RMSE(predict, Test$PRICE\_ECONOMY)

Warning message:

In log(x) : NaNs produced

> predict<- predict(linear.mod, Test)

> RMSE<- RMSE(predict, Test$PRICE\_ECONOMY)

Warning message:

In log(x) : NaNs produced

> RMSE

[1] NaN

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1,

+ importance = T

+ varImpPlot(model.forest)

Error: unexpected symbol in:

" importance = T

varImpPlot"

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1)

Error in randomForest(PRICE\_ECONOMY ~ PITCH\_ECONOMY + WIDTH\_ECONOMY + :

could not find function "randomForest"

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1)

Error in randomForest(PRICE\_ECONOMY ~ PITCH\_ECONOMY + WIDTH\_ECONOMY + :

could not find function "randomForest"

>

> varImpPlot(model.forest)

Error in varImpPlot(model.forest) : could not find function "varImpPlot"

> library(rpart)

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1)

Error in randomForest(PRICE\_ECONOMY ~ PITCH\_ECONOMY + WIDTH\_ECONOMY + :

could not find function "randomForest"

> install.packages("randomForest")

Installing package into ‘C:/Users/USER/Documents/R/win-library/3.4’

(as ‘lib’ is unspecified)

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/randomForest\_4.6-12.zip'

Content type 'application/zip' length 179239 bytes (175 KB)

downloaded 175 KB

package ‘randomForest’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in

C:\Users\USER\AppData\Local\Temp\Rtmp2HmsxM\downloaded\_packages

> library(rpart)

> library(randomForest)

randomForest 4.6-12

Type rfNews() to see new features/changes/bug fixes.

Attaching package: ‘randomForest’

The following object is masked from ‘package:ranger’:

importance

The following object is masked from ‘package:ggplot2’:

margin

The following object is masked from ‘package:psych’:

outlier

Warning message:

package ‘randomForest’ was built under R version 3.4.1

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1)

> varImpPlot(model.forest)

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1,

+ importance = T)

> varImpPlot(model.forest)

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_ECONOMY))^2))

> round(rmse, digits = 3)

[1] 0.327

> library(rpart)

> library(randomForest)

> model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1,

+ importance = T)

>

> varImpPlot(model.forest)

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_ECONOMY))^2))

> round(rmse, digits = 3)

[1] 0.323

> model <- rpart(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")

> predict <- predict(model, Test)

> RMSE1 <- RMSE(predict, Test$PRICE\_ECONOMY)

Error in RMSE(predict, Test$PRICE\_ECONOMY) :

could not find function "RMSE"

> RMSE <- function(x,y)

+ {

+ a <- sqrt(sum((log(x)-log(y))^2)/length(y))

+ return(a)

+ }

>

>

> model <- rpart(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")

> predict <- predict(model, Test)

> RMSE1 <- RMSE(predict, Test$PRICE\_ECONOMY)

> RMSE1 <- round(RMSE1, digits = 3)

> RMSE1

[1] 0.476

> linear.mod<- lm(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

> summary(linear.mod)

Call:

lm(formula = PRICE\_PREMIUM ~ PITCH\_PREMIUM + WIDTH\_PREMIUM +

FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

Residuals:

Min 1Q Median 3Q Max

-2073.5 -604.9 -170.4 760.1 4531.0

Coefficients:

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

(Intercept) -5499.49 4901.97 -1.122 0.263

PITCH\_PREMIUM 155.22 148.56 1.045 0.297

WIDTH\_PREMIUM 29.42 77.51 0.380 0.705

FLIGHT\_DURATION 215.60 16.59 12.995 <2e-16 \*\*\*

QUALITY -106.15 113.77 -0.933 0.351

PRICE\_RELATIVE -225.64 141.43 -1.595 0.112

---

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

Residual standard error: 964.8 on 341 degrees of freedom

Multiple R-squared: 0.4151, Adjusted R-squared: 0.4065

F-statistic: 48.4 on 5 and 341 DF, p-value: < 2.2e-16

> linear.mod<- lm(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

> summary(linear.mod)

Call:

lm(formula = PRICE\_ECONOMY ~ PITCH\_ECONOMY + WIDTH\_ECONOMY +

FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

Residuals:

Min 1Q Median 3Q Max

-1503.78 -485.28 -16.78 587.93 1655.51

Coefficients:

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

(Intercept) -8566.89 3343.91 -2.562 0.0108 \*

PITCH\_ECONOMY 574.45 100.45 5.719 2.35e-08 \*\*\*

WIDTH\_ECONOMY -580.68 69.37 -8.370 1.50e-15 \*\*\*

FLIGHT\_DURATION 178.61 11.96 14.934 < 2e-16 \*\*\*

QUALITY 200.08 35.49 5.637 3.63e-08 \*\*\*

PRICE\_RELATIVE -872.63 91.76 -9.510 < 2e-16 \*\*\*

---

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

Residual standard error: 649.2 on 341 degrees of freedom

Multiple R-squared: 0.569, Adjusted R-squared: 0.5626

F-statistic: 90.02 on 5 and 341 DF, p-value: < 2.2e-16

> linear.mod<- lm(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

> summary(linear.mod)

Call:

lm(formula = PRICE\_PREMIUM ~ PITCH\_PREMIUM + WIDTH\_PREMIUM +

FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

Residuals:

Min 1Q Median 3Q Max

-2073.5 -604.9 -170.4 760.1 4531.0

Coefficients:

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

(Intercept) -5499.49 4901.97 -1.122 0.263

PITCH\_PREMIUM 155.22 148.56 1.045 0.297

WIDTH\_PREMIUM 29.42 77.51 0.380 0.705

FLIGHT\_DURATION 215.60 16.59 12.995 <2e-16 \*\*\*

QUALITY -106.15 113.77 -0.933 0.351

PRICE\_RELATIVE -225.64 141.43 -1.595 0.112

---

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

Residual standard error: 964.8 on 341 degrees of freedom

Multiple R-squared: 0.4151, Adjusted R-squared: 0.4065

F-statistic: 48.4 on 5 and 341 DF, p-value: < 2.2e-16

> ibrary(rpart)

Error in ibrary(rpart) : could not find function "ibrary"

> library(randomForest)

> model.forest <- randomForest(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1,

+ importance = T)

>

> varImpPlot(model.forest)

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

+ round(rmse, digits = 3)

Error: unexpected symbol in:

"rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

round"

>

> # Evaluation metric function

> #A custom root mean Square Function to evaluate the performance of our model

> RMSE <- function(x,y)

+ {

+ a <- sqrt(sum((log(x)-log(y))^2)/length(y))

+ return(a)

+ }

>

> #Implementing the Regression Tree Model

> model <- rpart(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")

> predict <- predict(model, Test)

> RMSE1 <- RMSE(predict, Test$PRICE\_ECONOMY)

> RMSE1 <- round(RMSE1, digits = 3)

> RMSE1

[1] 0.476

>

>

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

+ round(rmse, digits = 3)

Error: unexpected symbol in:

"rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

round"

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

+ round(rmse, digits = 3)

Error: unexpected symbol in:

"rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

round"

>

> # Evaluation metric function

> #A custom root mean Square Function to evaluate the performance of our model

> RMSE <- function(x,y)

+ {

+ a <- sqrt(sum((log(x)-log(y))^2)/length(y))

+ return(a)

+ }

>

> #Implementing the Regression Tree Model

> model <- rpart(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")

> predict <- predict(model, Test)

> RMSE1 <- RMSE(predict, Test$PRICE\_ECONOMY)

> RMSE1 <- round(RMSE1, digits = 3)

> RMSE1

[1] 0.476

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

+ round(rmse, digits = 3)

Error: unexpected symbol in:

"rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

round"

>

> # Evaluation metric function

> #A custom root mean Square Function to evaluate the performance of our model

> RMSE <- function(x,y)

+ {

+ a <- sqrt(sum((log(x)-log(y))^2)/length(y))

+ return(a)

+ }

>

> #Implementing the Regression Tree Model

> model <- rpart(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")

> predict <- predict(model, Test)

> RMSE1 <- RMSE(predict, Test$PRICE\_ECONOMY)

> RMSE1 <- round(RMSE1, digits = 3)

> RMSE1

[1] 0.476

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_ECONOMY))^2))

> round(rmse, digits = 3)

[1] 0.564

> prediction <- predict(model.forest,Test)

> rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

+ round(rmse, digits = 3)

Error: unexpected symbol in:

"rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

round"

> model.forest <- randomForest(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",

+ ntree = 300,

+ mtry = 2, #mtry is sqrt(6)

+ replace = F,

+ nodesize = 1,

+ importance = T)

> prediction <- predict(model.forest,Test)

> rmse2 <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

+ round(rmse2, digits = 3)

Error: unexpected symbol in:

"rmse2 <- sqrt(mean((log(prediction)-log(Test$PRICE\_PREMIUM)^2))

round"

> AIC(linear.mod)

[1] 5761.775

> BIC(linear.mod)

[1] 5788.72

> AIC(linear.mod1)

Error in AIC(linear.mod1) : object 'linear.mod1' not found

> linear.mod1<- lm(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + PRICE\_RELATIVE, data = Training)

> AIC(linear.mod1)

[1] 5713.11

> BIC(linear.mod1)

[1] 5732.357

> predict<- predict(linear.mod, Test)

> actuals\_preds <- data.frame(cbind(actuals=Test$PRICE\_ECONOMY, predicteds=predict)) # make actuals\_predicteds dataframe.

> correlation\_accuracy <- cor(actuals\_preds)

> correlation\_accuracy

actuals predicteds

actuals 1.0000000 0.6859838

predicteds 0.6859838 1.0000000

> min\_max\_accuracy <- mean (apply(actuals\_preds, 1, min) / apply(actuals\_preds, 1, max))

> min\_max\_accuracy

[1] 0.6309309

> mape <- mean(abs((actuals\_preds$predicteds - actuals\_preds$actuals))/actuals\_preds$actuals)

> mape

[1] 1.089928

> data.df <-read.csv(paste("C:/Users/USER/Downloads/SixAirlines.csv", sep=""))#Loading the Data set

> attach(data.df)#Attaching the Data ste

The following objects are masked from data.df (pos = 11):

AIRCRAFT, AIRLINE, FLIGHT\_DURATION, INTERNATIONAL, LAMBDA, MONTH, N, PITCH\_ECONOMY,

PITCH\_PREMIUM, PRICE\_ECONOMY, PRICE\_PREMIUM, PRICE\_RELATIVE, QUALITY, SEATS\_ECONOMY,

SEATS\_PREMIUM, WIDTH\_ECONOMY, WIDTH\_PREMIUM

The following objects are masked from data.df (pos = 12):

AIRCRAFT, AIRLINE, FLIGHT\_DURATION, INTERNATIONAL, LAMBDA, MONTH, N, PITCH\_ECONOMY,

PITCH\_PREMIUM, PRICE\_ECONOMY, PRICE\_PREMIUM, PRICE\_RELATIVE, QUALITY, SEATS\_ECONOMY,

SEATS\_PREMIUM, WIDTH\_ECONOMY, WIDTH\_PREMIUM

The following objects are masked from data.df (pos = 13):

AIRCRAFT, AIRLINE, FLIGHT\_DURATION, INTERNATIONAL, LAMBDA, MONTH, N, PITCH\_ECONOMY,

PITCH\_PREMIUM, PRICE\_ECONOMY, PRICE\_PREMIUM, PRICE\_RELATIVE, QUALITY, SEATS\_ECONOMY,

SEATS\_PREMIUM, WIDTH\_ECONOMY, WIDTH\_PREMIUM

> View(data.df)#General view of the entire Data frme

> #Simple Statistical Analysis using describe funtion

> library(psych)

> describe(data.df)

vars n mean sd median trimmed mad min max range skew kurtosis se

AIRLINE\* 1 462 3.02 1.65 2.00 2.90 1.48 1.00 6.00 5.00 0.59 -0.95 0.08

AIRCRAFT 2 462 0.33 0.47 0.00 0.28 0.00 0.00 1.00 1.00 0.74 -1.46 0.02

FLIGHT\_DURATION 3 462 7.55 3.54 7.75 7.54 4.82 1.25 14.66 13.41 -0.05 -1.12 0.16

MONTH 4 462 1.67 1.05 2.00 1.71 1.48 0.00 3.00 3.00 -0.16 -1.20 0.05

INTERNATIONAL 5 462 0.91 0.28 1.00 1.00 0.00 0.00 1.00 1.00 -2.93 6.60 0.01

SEATS\_ECONOMY 6 462 200.71 77.96 185.00 193.76 85.99 17.00 389.00 372.00 0.61 -0.26 3.63

SEATS\_PREMIUM 7 462 33.54 13.26 36.00 33.20 11.86 8.00 66.00 58.00 0.25 -0.46 0.62

PITCH\_ECONOMY 8 462 31.21 0.66 31.00 31.25 0.00 30.00 33.00 3.00 -0.03 -0.38 0.03

PITCH\_PREMIUM 9 462 37.92 1.32 38.00 38.06 0.00 34.00 40.00 6.00 -1.48 3.43 0.06

WIDTH\_ECONOMY 10 462 17.83 0.56 18.00 17.81 0.00 17.00 19.00 2.00 -0.03 -0.12 0.03

WIDTH\_PREMIUM 11 462 19.48 1.10 19.00 19.54 0.00 17.00 21.00 4.00 -0.09 -0.34 0.05

PRICE\_ECONOMY 12 462 1317.06 989.81 1224.00 1231.30 1163.84 65.00 3593.00 3528.00 0.52 -0.88 46.05

PRICE\_PREMIUM 13 462 1832.35 1289.97 1710.00 1782.94 1852.51 86.00 7414.00 7328.00 0.51 0.41 60.01

PRICE\_RELATIVE 14 462 0.49 0.45 0.38 0.43 0.42 0.02 1.89 1.87 1.14 0.61 0.02

N 15 462 234.25 86.88 227.00 227.69 90.44 38.00 441.00 403.00 0.61 -0.44 4.04

LAMBDA 16 462 0.15 0.06 0.13 0.14 0.03 0.05 0.55 0.50 2.70 14.02 0.00

QUALITY 17 462 6.72 1.78 7.00 6.79 0.00 2.00 10.00 8.00 -0.51 1.67 0.08

> library(ggplot2)

> ## Loading required package: ggplot2

> #Seggregating different flights

> ggplot(data.df, aes(x = AIRLINE, fill = AIRLINE)) + geom\_bar()

> #Seggregating international and domestic flights

> ggplot(data.df, aes(x = INTERNATIONAL))+ geom\_bar()

> #Prices of Economy and Premium tickets

> ggplot(data.df, aes(x = PRICE\_ECONOMY)) + geom\_density()

> ggplot(data.df, aes(x = PRICE\_PREMIUM)) + geom\_density()

> # A Scatterplot of price economy vs flight hours of travel

> # ==========

> plot(FLIGHT\_DURATION,PRICE\_ECONOMY,

+ col="blue",

+ main="Price economy vs flight hours",

+ xlab="Hours", ylab="Price")

>

> # Add the sample means to the Scatterplot

> # ==========

> abline(h=mean(PRICE\_ECONOMY), col="dark blue", lty="dotted")

> abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

>

> # Add a regression line

> # ==========

> abline(lm(PRICE\_ECONOMY ~ FLIGHT\_DURATION))

> # A Scatterplot of price premium vs flight hours of travel

> # ==========

> plot(FLIGHT\_DURATION,PRICE\_PREMIUM,

+ col="blue",

+ main="Price economy vs flight hours",

+ xlab="Hours", ylab="Price")

>

> # Add the sample means to the Scatterplot

> # ==========

> abline(h=mean(PRICE\_PREMIUM), col="dark blue", lty="dotted")

> abline(v=mean(FLIGHT\_DURATION), col="dark blue", lty="dotted")

>

> # Add a regression line

> # ==========

> abline(lm(PRICE\_PREMIUM ~ FLIGHT\_DURATION))

> #Correlation and Correlation Matrix for Price Economy

>

> library(corrplot)

> library(gplots) # for color interpolation

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

> corrplot.mixed(corr=cor(data.df[ , c(2:6, 8,10,12,15,17)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

> #Correlation and Correlation Matrix for Price Premium

>

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

> corrplot.mixed(corr=cor(data.df[ , c(2:6, 7,9,11,15,17)], use="complete.obs"),

+ upper="ellipse", tl.pos="lt",

+ col = colorpanel(50, "red", "gray60", "blue4"))

> #Scatter Plot Matrix for Price Economy and Price Premium

> library(car)

> scatterplotMatrix(formula = ~ SEATS\_ECONOMY + PITCH\_ECONOMY + WIDTH\_ECONOMY + PRICE\_ECONOMY, cex=0.6,

+ data=data.df, diagonal="histogram")

Warning message:

In smoother(x, y, col = col[2], log.x = FALSE, log.y = FALSE, spread = spread, :

could not fit smooth

> scatterplotMatrix(formula = ~ SEATS\_PREMIUM + PITCH\_PREMIUM + WIDTH\_PREMIUM + PRICE\_PREMIUM, cex=0.6,

+ data=data.df, diagonal="histogram")

Warning message:

In smoother(x, y, col = col[2], log.x = FALSE, log.y = FALSE, spread = spread, :

could not fit smooth

> #Calculating correlations between Prices of Economy and Premium in correlation to other factors

> cor.test(PRICE\_ECONOMY, PITCH\_ECONOMY)

Pearson's product-moment correlation

data: PRICE\_ECONOMY and PITCH\_ECONOMY

t = 8.8003, df = 460, p-value < 2.2e-16

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.2987210 0.4550742

sample estimates:

cor

0.379605

> cor.test(PRICE\_ECONOMY, WIDTH\_ECONOMY)

Pearson's product-moment correlation

data: PRICE\_ECONOMY and WIDTH\_ECONOMY

t = 1.764, df = 460, p-value = 0.0784

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.009330795 0.171911298

sample estimates:

cor

0.0819679

> cor.test(PRICE\_PREMIUM, PITCH\_PREMIUM)

Pearson's product-moment correlation

data: PRICE\_PREMIUM and PITCH\_PREMIUM

t = 1.5338, df = 460, p-value = 0.1258

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.02002801 0.16150915

sample estimates:

cor

0.07133125

> cor.test(PRICE\_PREMIUM, WIDTH\_PREMIUM)

Pearson's product-moment correlation

data: PRICE\_PREMIUM and WIDTH\_PREMIUM

t = 1.0592, df = 460, p-value = 0.2901

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

-0.04209336 0.13992426

sample estimates:

cor

0.04932498

*CODE :*

*#Using the Boruta package to calcuate the effectiveness of different variables in Calculating the price of Economy class tickets*

*eco.df <-read.csv(paste("C:/Users/USER/Downloads/Economy.csv", sep=""))*

*library(Boruta)*

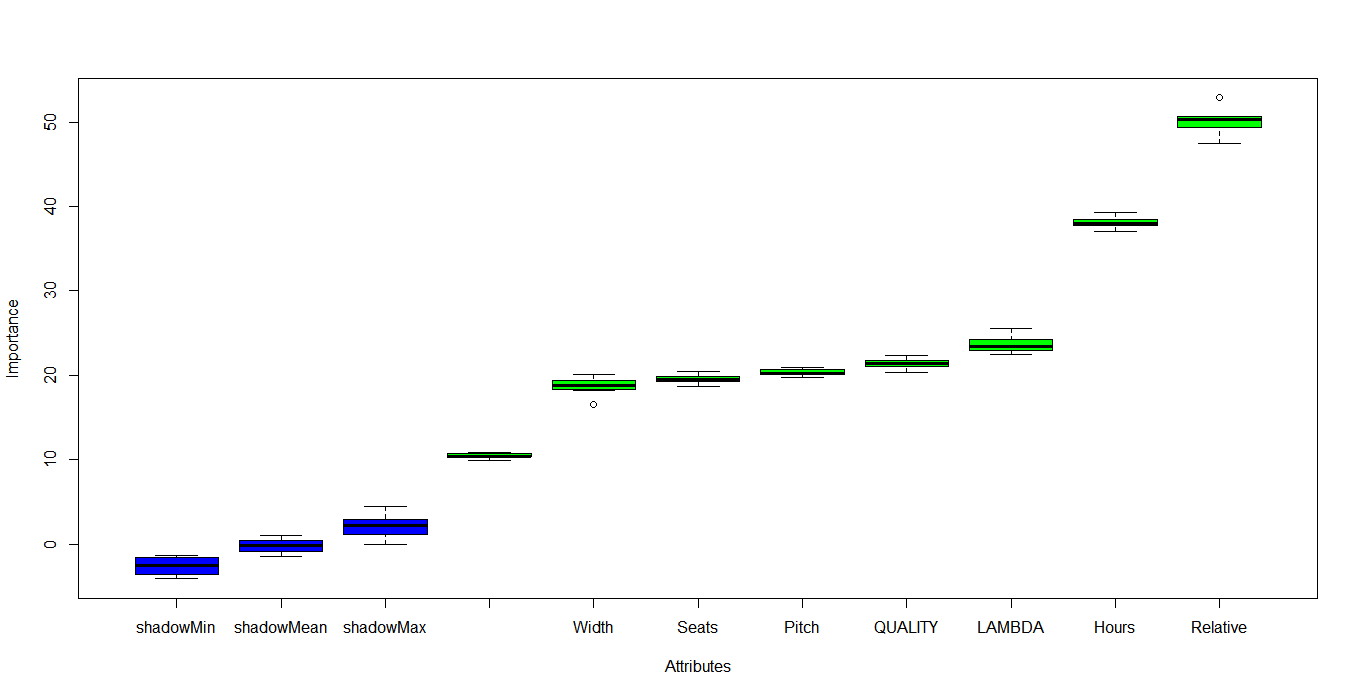
*set.seed(1234) # for code reproducibility*

*response <- data.df$PRICE\_ECONOMY*

*bor.results <- Boruta(eco.df,response,maxRuns=101,doTrace=0)*

*plot(bor.results)*

*OUTPUT :*

**

*CODE :*

*#Using the Boruta package to calcuate the effectiveness of different variables in Calculating the price of Premium class tickets*

*pre.df <-read.csv(paste("C:/Users/USER/Downloads/prem.csv", sep=""))*

*library(Boruta)*

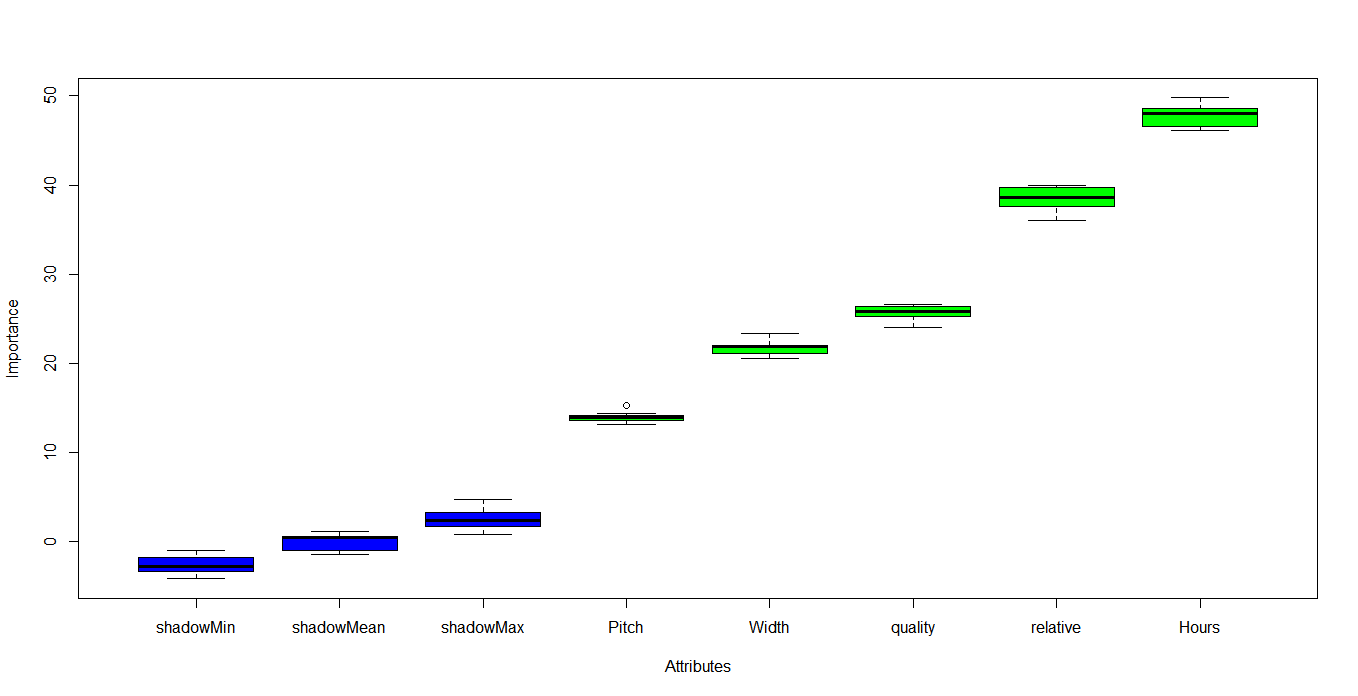
*set.seed(1234) # for code reproducibility*

*response <- data.df$PRICE\_PREMIUM*

*bor.results <- Boruta(pre.df,response,maxRuns=101,doTrace=0)*

*plot(bor.results)*

*OUTPUT :*

**

*CODE :*

*#Dividing the Data set into Test and Training Data ste*

*ratio = sample(1:nrow(data.df), size = 0.25\*nrow(data.df))*

*Test = data.df[ratio,] #Test dataset 25% of total*

*Training = data.df[-ratio,] #Train dataset 75% of total*

*dim(Training)*

*dim(Test)*

*OUTPUT :*

|  |
| --- |
| dim(Training)  [1] 347 17  > dim(Test)  [1] 115 17 |
|  |
| |  | | --- | |  | |

*CODE :*

*#Generating A Multi Variable Linear Regressional Model for Price of Economy Flights*

*linear.mod<- lm(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)*

*summary(linear.mod)*

*OUTPUT :*

Call:

lm(formula = PRICE\_ECONOMY ~ PITCH\_ECONOMY + WIDTH\_ECONOMY +

FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

Residuals:

Min 1Q Median 3Q Max

-1503.78 -485.28 -16.78 587.93 1655.51

Coefficients:

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

(Intercept) -8566.89 3343.91 -2.562 0.0108 \*

PITCH\_ECONOMY 574.45 100.45 5.719 2.35e-08 \*\*\*

WIDTH\_ECONOMY -580.68 69.37 -8.370 1.50e-15 \*\*\*

FLIGHT\_DURATION 178.61 11.96 14.934 < 2e-16 \*\*\*

QUALITY 200.08 35.49 5.637 3.63e-08 \*\*\*

PRICE\_RELATIVE -872.63 91.76 -9.510 < 2e-16 \*\*\*

---

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

Residual standard error: 649.2 on 341 degrees of freedom

Multiple R-squared: 0.569, Adjusted R-squared: 0.5626

F-statistic: 90.02 on 5 and 341 DF, p-value: < 2.2e-16

*CODE :*

*#Calculating AIC and BIC for the Linear Model*

*AIC(linear.mod)*

*BIC(linear.mod)*

*OUTPUT :*

> AIC(linear.mod)

[1] 5486.814

> BIC(linear.mod)

[1] 5513.76

*CODE :*

*#Changing the model a bit by eliminating Quaity and Flight Duration*

*linear.mod1<- lm(PRIE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + PRICE\_RELATIVE, data = Training)*

*AIC(linear.mod1)*

*BIC(linear.mod1)*

*#The second sets of Models are considered to be better*

*OUTPUT :*

AIC(linear.mod1)

[1] 5713.11

> BIC(linear.mod1)

[1] 5732.357

> #The second sets of Models are considered to be better

*CODE :*

*library(rpart)*

*library(randomForest)*

*model.forest <- randomForest(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",*

*ntree = 300,*

*mtry = 2, #mtry is sqrt(6)*

*replace = F,*

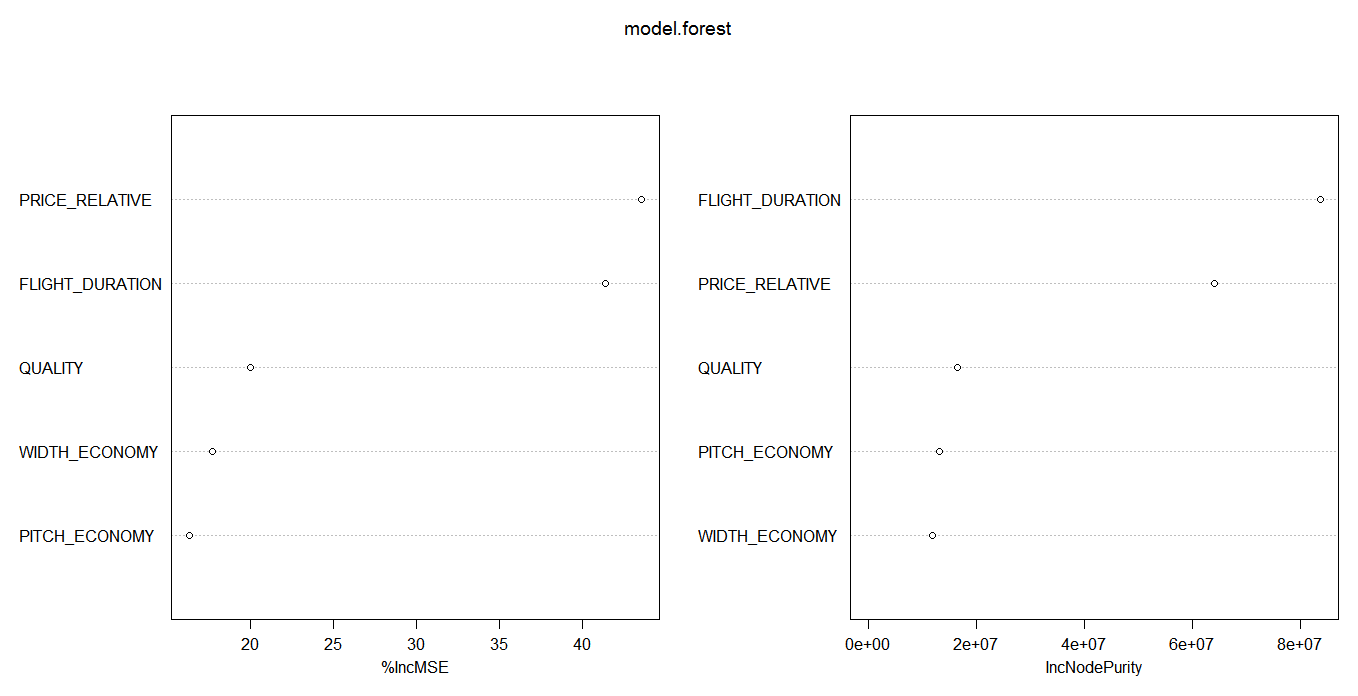
*nodesize = 1,*

*importance = T)*

*varImpPlot(model.forest)*

*#From the VIF plot we see that Flight Duration and Price Relative are most important factors in predicitng Price Economy.*

*OUTPUT :*

**

*CODE :*

*#We test the model using Random Forest*

*prediction <- predict(model.forest,Test)*

*rmse <- sqrt(mean((log(prediction)-log(Test$PRICE\_ECONOMY))^2))*

*round(rmse, digits = 3)*

*OUTPUT :*

*0.322*

*CODE :*

*# Evaluation metric function*

*#A custom root mean Square Function to evaluate the performance of our model*

*RMSE <- function(x,y)*

*{*

*a <- sqrt(sum((log(x)-log(y))^2)/length(y))*

*return(a)*

*}*

*#Implementing the Regression Tree Model*

*model <- rpart(PRICE\_ECONOMY~ PITCH\_ECONOMY + WIDTH\_ECONOMY + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")*

*predict <- predict(model, Test)*

*RMSE1 <- RMSE(predict, Test$PRICE\_ECONOMY)*

*RMSE1 <- round(RMSE1, digits = 3)*

*RMSE1*

*OUTPUT :*

*0.476*

*CODE :*

*#PREDICTING THE ACCURACY OF THE MODEL*

*predict<- predict(linear.mod, Test)*

*actuals\_preds <- data.frame(cbind(actuals=Test$PRICE\_ECONOMY, predicteds=predict)) # make actuals\_predicteds dataframe.*

*correlation\_accuracy <- cor(actuals\_preds)*

*correlation\_accuracy*

*OUTPUT :*

actuals predicteds

actuals 1.0000000 0.7866478

predicteds 0.7866478 1.0000000

*CODE :*

*#CALCULATING THE MIN MAX ACCURACY AND MAPE*

*min\_max\_accuracy <- mean (apply(actuals\_preds, 1, min) / apply(actuals\_preds, 1, max))*

*min\_max\_accuracy*

*mape <- mean(abs((actuals\_preds$predicteds - actuals\_preds$actuals))/actuals\_preds$actuals)*

*mape*

*OUTPUT :*

> min\_max\_accuracy

[1] 0.5719264

>

> mape <- mean(abs((actuals\_preds$predicteds - actuals\_preds$actuals))/actuals\_preds$actuals)

> mape

[1] 0.7390665

*CODE :*

*#For Premium Class Tickets*

*#Generating A Multi Variable Linear Regressional Model for Price of Premium Flights*

*linear.mod<- lm(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)*

*summary(linear.mod)*

*#The model has an F Statistic of 48.4 which is mediumly high*

*#the t value of Pitch\_premium, width\_premium, Price\_relative and quality is positive indicating that these predictors are associated with*

*#Price\_Premium. A larger t-value indicates that that it is less likely that the coefficient is not equal to zero purely by chance.*

*#Again, as the p-value for Flight\_Duration is less than 0.05 they are both statistically significant in the multiple linear regression model for Price\_Economy response variable.*

*#The model’s, p-value: < 2.2e-16 is also lower than the statistical significance level of 0.05, this indicates that we can safely reject the null hypothesis that the value for the coefficient is zero*

*#(or in other words, the predictor variable has no explanatory relationship with the response variable).*

*OUTPUT :*

Call:

lm(formula = PRICE\_PREMIUM ~ PITCH\_PREMIUM + WIDTH\_PREMIUM +

FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training)

Residuals:

Min 1Q Median 3Q Max

-2073.5 -604.9 -170.4 760.1 4531.0

Coefficients:

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

(Intercept) -5499.49 4901.97 -1.122 0.263

PITCH\_PREMIUM 155.22 148.56 1.045 0.297

WIDTH\_PREMIUM 29.42 77.51 0.380 0.705

FLIGHT\_DURATION 215.60 16.59 12.995 <2e-16 \*\*\*

QUALITY -106.15 113.77 -0.933 0.351

PRICE\_RELATIVE -225.64 141.43 -1.595 0.112

---

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

Residual standard error: 964.8 on 341 degrees of freedom

Multiple R-squared: 0.4151, Adjusted R-squared: 0.4065

F-statistic: 48.4 on 5 and 341 DF, p-value: < 2.2e-16

*CODE :*

*library(rpart)*

*library(randomForest)*

*model.forest <- randomForest(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova",*

*ntree = 300,*

*mtry = 2, #mtry is sqrt(6)*

*replace = F,*

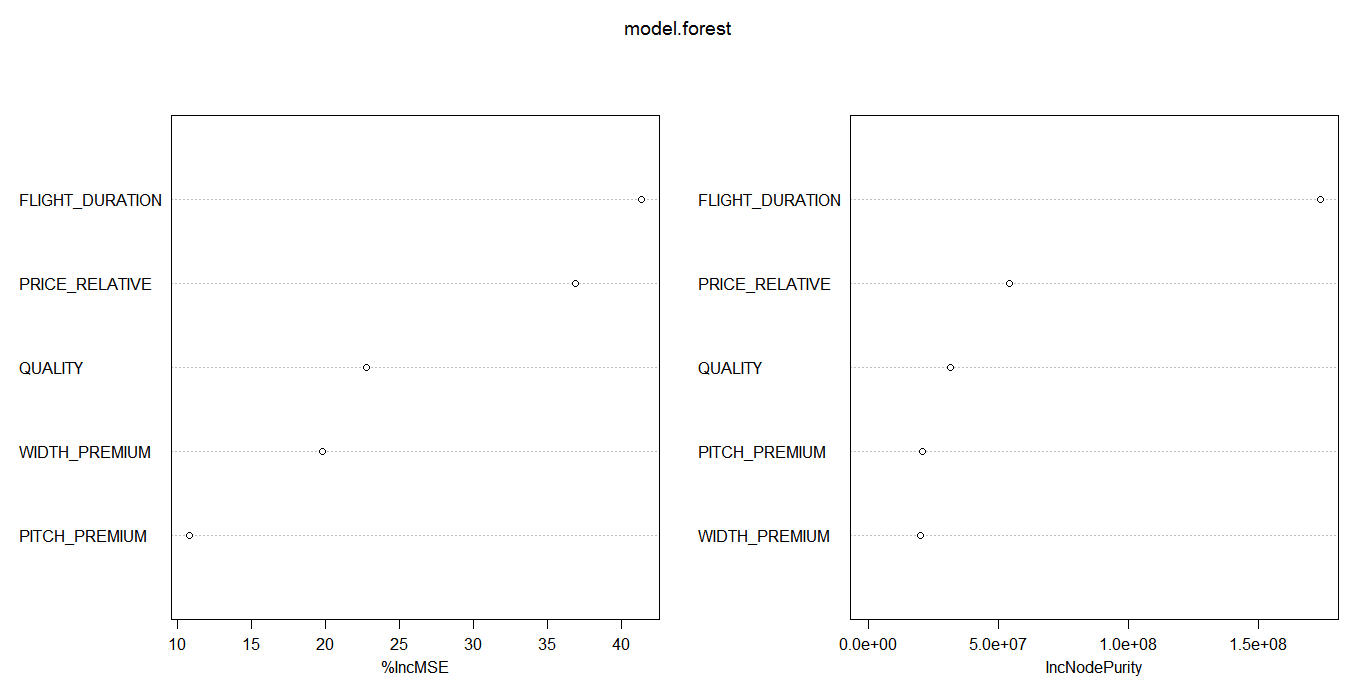
*nodesize = 1,*

*importance = T)*

*varImpPlot(model.forest)*

*#From the VIF plot we see that Flight Duration and Price Relative are most important factors in predicitng Price Economy.*

*OUTPUT :*

**

*CODE :*

*# Evaluation metric function*

*#A custom root mean Square Function to evaluate the performance of our model*

*RMSE <- function(x,y)*

*{*

*a <- sqrt(sum((log(x)-log(y))^2)/length(y))*

*return(a)*

*}*

*#Implementing the Regression Tree Model*

*model <- rpart(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + FLIGHT\_DURATION + QUALITY + PRICE\_RELATIVE, data = Training, method = "anova")*

*predict <- predict(model, Test)*

*RMSE1 <- RMSE(predict, Test$PRICE\_PREMIUM)*

*RMSE1 <- round(RMSE1, digits = 3)*

*RMSE1*

*OUTPUT :*

0.447

*CODE :*

*#Calculating AIC and BIC for the Linear Model*

*AIC(linear.mod)*

*BIC(linear.mod)*

*OUTPUT :*

AIC(linear.mod)

[1] 5761.775

> BIC(linear.mod)

[1] 5788.72

*CODE :*

*#Changing the model a bit by eliminating Quaity and Flight Duration*

*linear.mod1<- lm(PRICE\_PREMIUM~ PITCH\_PREMIUM + WIDTH\_PREMIUM + PRICE\_RELATIVE, data = Training)*

*AIC(linear.mod1)*

*BIC(linear.mod1)*

*#The FIRST sets of Models are considered to be better*

*OUTPUT :*

AIC(linear.mod1)

[1] 5942.036

> BIC(linear.mod1)

[1] 5961.283