```
library(dplyr)
library(lubridate)
library(stringr)
library(DataCombine)
library(corrgram)
library(psych)
library(forecast)
library(dendextend)
library(ROCR)
library(tree)
library(arules)
library(arulesViz)
library(gmodels)

library(stringr)
library(dplyr)
library(psych)
library(corrgram)
library(DataCombine)
corr.test(data[,2:6])
corrgram(data[ ,1:6], order = T, main = "Correlogram
of Marketing Data, Ordered", lower.panel =
panel.shade, upper.panel = panel.pie, diag.panel =
panel.minmax, text.panel = panel.txt)
current = unique(str_subset(bike$sources,
web_sites))
current
replace = rep("web", length(current))
replace
replacements = data.frame(from = current, to =
replace)
replacements
bike = FindReplace(data = bike, Var = "sources",
replacements, from = "from", to = "to", exact =
FALSE)

pairs(data)
model=lm(y~x,data=df)
summary(model)
plot(model$x,model$y,main='Linearity?')
qqnorm(model$residuals,main='Normality?')
qqline(model$residuals)
plot(model$fitted.values,model$residuals,
main='Homoscedasticity')
library(MASS)
boxcox(model)
lm(y~log(x),data=df)
lm(y^2~x,data=df)
newdata = data.frame(x=c(450,460,470))
predict.lm(m1, newdata, interval = 'predict')
predict.lm(m1, newdata, interval = 'confidence')

library(lubridate)
library(forecast)
plot(decompose(data))
as.Date(data$datetime,format='%y/%m/%d')
data%>%group_by(year=year(data$datetime,
month=month(data$datetime))%>%
summarise(x=sum(x))
table(data$year,data$month)
year(data$datetime);month(data$datetime)
ts(data,frequency=12,start=c(2011,1))
diff(diff(data),lag=12)
acf(data);pacf(data)
model=arima(data, c(1,0,0),
seasonal=list(order=c(0,0,0)))
tsdiag(model)
auto.arima(data)
plot(forecast(auto.arima(data)),h=12)
plot(forecast(tbats(data)),h=12)
hist(residuals(model))

library(dendextend)
model=kmeans(data[,c(2,3)], 3)
plot(x,y,col=model$cluster)
as.numeric(scale(data$x))
as.numeric(scale(data$y))
cor.test(x,y)
as.numeric(model$tot.withinss)
plot(data$x,data$y,col=model$cluster)

hc=hclust(dist(market[,4:5]),method='ward.D2')
dend=as.dendrogram(hc)
dend_col=color_branches(dend,k=6)
plot(dend_col,leaflab='none',horiz=T)
str(cut(dend,h=37.5)$upper)
dend5=cutree(dend,k=5)
plot(data$x,data$y,col=dend5))

income_mean <- attr(market$income_scale,
"scaled:center")
income_sd <- attr(market$income_scale,
"scaled:scale")
market$income_unscaled <- market$income_scale
* income_sd +
```

```
hc_mod = hclust(dist(market[,4:5]), method =
'ward.D2')
dend = as.dendrogram(hc_mod)
dend_six_color = color_branches(dend, k = 6)
plot(dend_six_color, leaflab = "none", horiz = F,
main = 'Age and Income Dendogram', ylab =
'Height')
abline(h = 37.5, lty= 'dashed', col = 'blue')
str(cut(dend, h =37.5)$upper)
optimize = data.frame(
clusters = c(2:10),
wss = rep(0,9)
)
for (i in seq(2, 10, by=1)) {
x = kmeans(market[,4:5], i)
optimize[i-1,2] = as.numeric(x$tot.withinss)
}
plot(optimize$wss ~ optimize$clusters, type = 'b',
main = 'Finding optimal number of clusters based
on error',
xlab = 'Number of Clusters',
ylab = 'Within Sum of Squared Error')

dend_six = cutree(dend, k = 6)
market$dend6 = dend_six

market %>% group_by(dend6) %>%
summarise(
Age_Range = paste(min(age),'-',max(age)),
Age_Median = median(age),
Income_range =
paste(round(min(income),2),'-',round(max(income),2)
),
Income_Median = median(income)
) %>%
mutate(Label = custom_labels)

library(ROCR)
library(gmodels)
library(tree)
indexes = sample(1:1000, size = 500)
Train = data[indexes,]
Test = data[-indexes,]

logisticmodel50final  <- glm(formula = Creditability
~ Account.Balance +
Payment.Status.of.Previous.Credit + Purpose +
Length.of.current.employment +
Sex...Marital.Status,family = "binomial", data =
Train)
fit50 = fitted.values(logisticmodel50final)
thres = rep(0,500)
for (i in 1:500) {
if(fit50[i]>0.5) {
   thres[i] = 1
} else {
   thres[i] = 0
   }
}
str(thres)
conf.mat = table(Train$Creditability, thres)
LR_train_acc = sum(diag(conf.mat))/500*100

ct = CrossTable(Test$Creditability, thres_pred,
digits=1, prop.r=F, prop.t=F,prop.chisq = F, chisq =
F)
print(ct)

prod_pred = prediction(fit50, Train$Creditability)
perf = performance(prod_pred,'tpr','fpr')
plot(perf, main='ROC-AUC Curve Training
Data');abline(a = 0, b = 1, col = "red", lty = 2)

tree_model = tree(Creditability ~ Account.Balance,
data=Train, method="class")
plot(tree_model);text(tree_model, pretty=0, cex=0.6)
train_pred = predict(tree_model,Train, type='class')
ct1 = table(Train$Creditability, train_pred)
T_Train_acc = sum(diag(ct1))/500*100
fit_tree_train = predict(tree_model, data=Train)
prod_pred = prediction(fit_tree_train[,2],
Train$Creditability)
perf1 = performance(prod_pred,'tpr','fpr')
plot(perf1, main='ROC-AUC Curve Training
Data');abline(a = 0, b = 1, col = "red", lty = 2)

tree_model_prune = prune.misclass(tree_model,
best=8)

library(arules)
library(arulesViz)
rules = apriori(Groceries, parameter = list(support =
0.05, confidence = 0.1))
sorted_rules = sort(rules, by='lift', decreasing=T)
inspect(sorted_rules)
plot(sorted_rules, method='graph')
plot(sorted_rules, method=’paracoord')
itemFrequencyPlot(Groceries, topN=10)
```