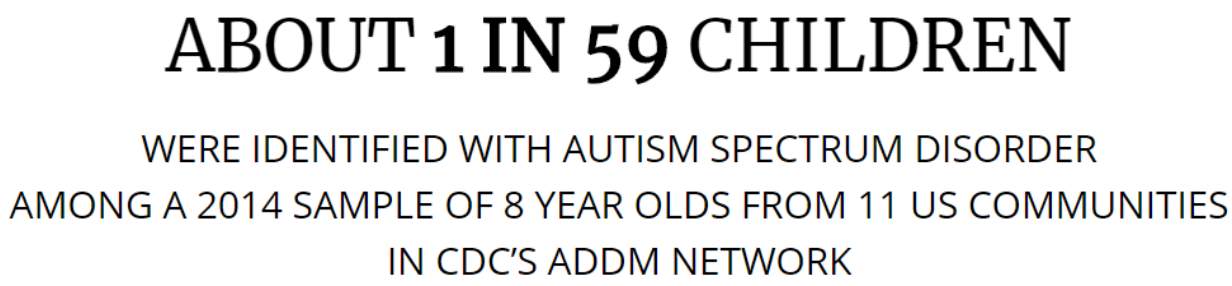
(https://github.com/dd-consulting)

One-Stop Analytics: Predictive Modeling (Rattle)

Case Study of Autism Spectrum Disorder (ASD) with R



[United States]

Centers for Disease Control and Prevention (CDC) - Autism Spectrum Disorder (ASD)

Autism spectrum disorder (ASD) is a developmental disability that can cause significant social, communication and behavioral challenges. CDC is committed to continuing to provide essential data on ASD, search for factors that put children at risk for ASD and possible causes, and develop resources that help identify children with ASD as early as possible.

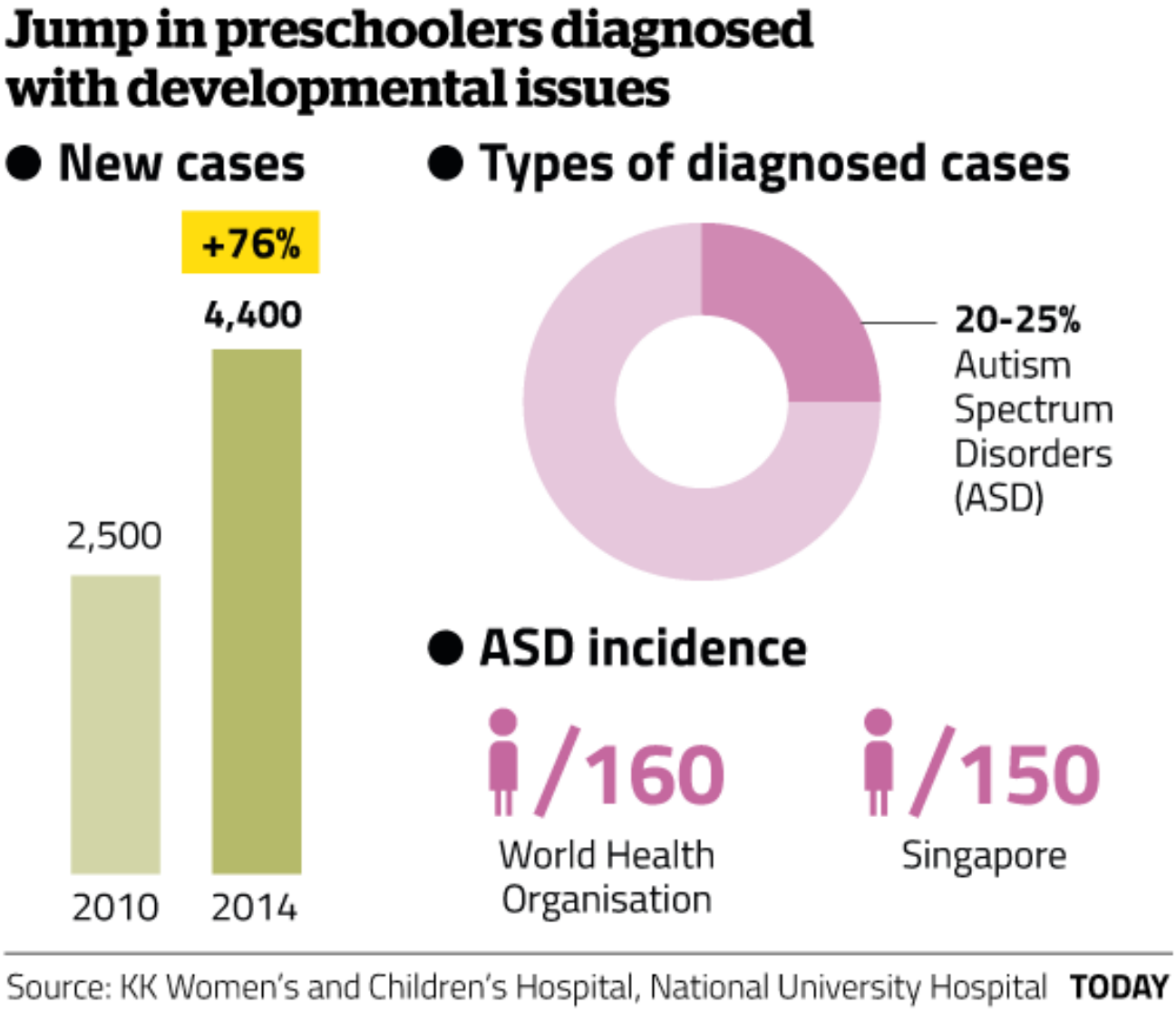
https://www.cdc.gov/ncbddd/autism/data/index.html (https://www.cdc.gov/ncbddd/autism/data/index.html)

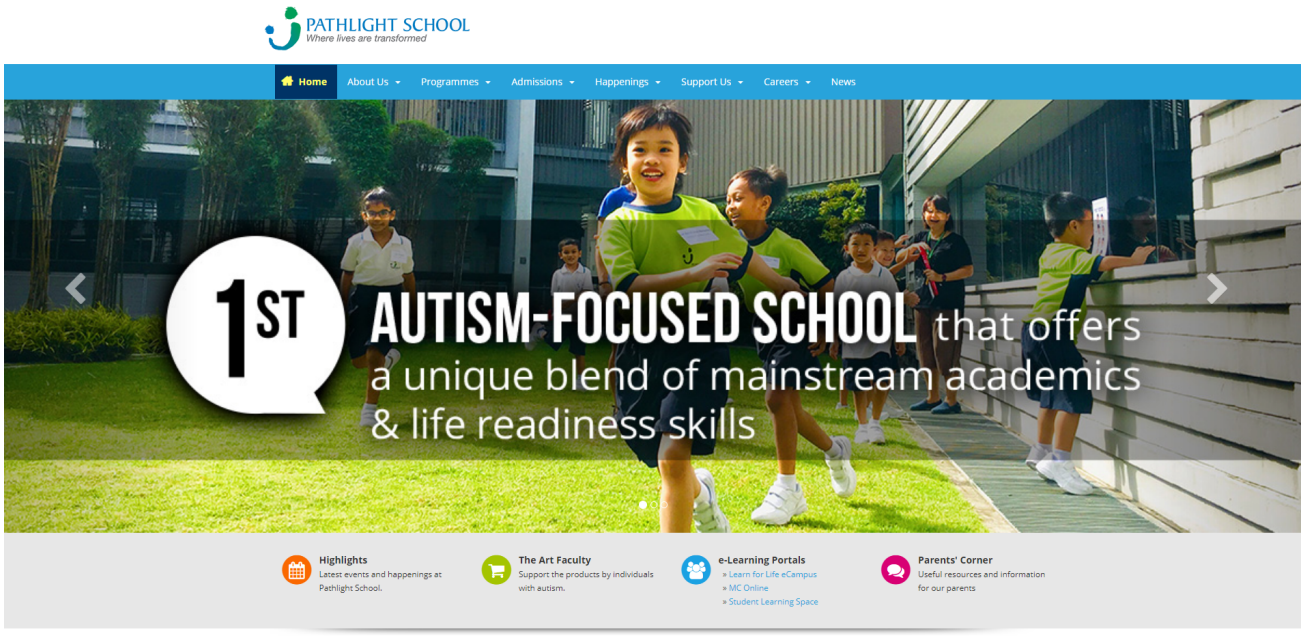
[Singapore]

TODAY Online - More preschoolers diagnosed with developmental issues

Doctors cited better awareness among parents and preschool teachers, leading to early referrals for diagnosis.

https://www.gov.sg/news/content/today-online-more-preschoolers-diagnosed-with-developmental-issues (https://www.gov.sg/news/content/today-online-more-preschoolers-diagnosed-with-developmental-issues)





https://www.pathlight.org.sg/ (https://www.pathlight.org.sg/)

()

Workshop Objective:

Use R to predict Autism Spectrum Disorder (ASD) prevalence.

https://www.cdc.gov/ncbddd/autism/data/index.html (https://www.cdc.gov/ncbddd/autism/data/index.html)

- Rattle
- Rattle: Import Data
- Rattle: EDA Explore & Test
- Rattle: Process & Transform Data

  Predict Prevelance Risk Levels (Classification)
- Rattle: Train Model (Classification)
  - Multi-Class Model: Decision Tree (DT)
  - Multi-Class Model: Random Forest (RF)
  - Multi-Class Model: Support Vector Machines (SVM)
  - Multi-Class Model: Multinomial Logistic Regression (MLR)
  - Binary-Class Model: Boost (AdaBoost & XgBoost)
  - Binary-Class Model: Neural Net (NN)
- Rattle: Evaluate Model (Classification)

  Predict Prevelance (Regression)
- Rattle: Train Model (Regression)
  - Regression Model: Decision Tree (DT)
  - Regression Model: Random Forest (RF)
  - Regression Model: Linear Regression (LR)
  - Regressions Model: Neural Net (NN)
- Rattle: Evaluate Model (Regression)
- Rattle: Improve Model
- Rattle: Save Model & Log
- Workshop Submission
- Appendices

()

In [1]:
```
library("repr") # Show graphs in-line notebook
```

Obtain current R working directory

In [2]:
```
getwd()
```

'/media/sf_vm_shared_folder/git/DDC/DDC-ASD/model_R'

Set new R working directory

In [3]:
```
# setwd("/media/sf_vm_shared_folder/git/DDC/DDC-ASD/model_R")
# setwd('~/Desktop/admin-desktop/vm_shared_folder/git/DDC-ASD/model_R')
getwd()
```

'/media/sf_vm_shared_folder/git/DDC/DDC-ASD/model_R'

In [4]:
```
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=5)
```

()

Rattle

Rattle — the **R** **A**nalytical **T**ool **T**o **L**earn **E**asily — is a popular open-source GUI for data mining using R.

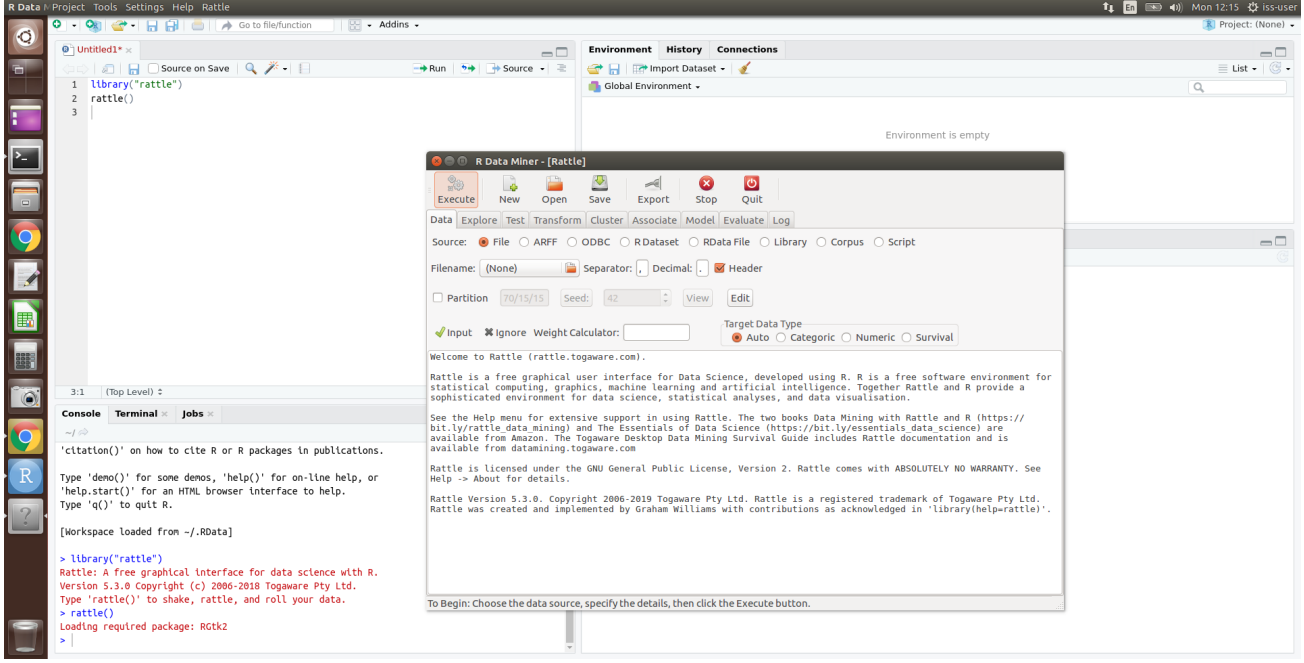
https://rattle.togaware.com/ (https://rattle.togaware.com/)

https://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf (https://journal.r-project.org/archive/2009-2/RJournal_2009-2_Williams.pdf)

Rattle

In [5]:
```
if(!require(rattle)){install.packages("rattle")}
library('rattle')
```

```
Loading required package: rattle
Rattle: A free graphical interface for data science with R.
Version 5.3.0 Copyright (c) 2006-2018 Togaware Pty Ltd.
Type 'rattle()' to shake, rattle, and roll your data.
```


R Data Miner - [Rattle]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Source: File ARFF ODBC R Dataset RData File Library Corpus Script
Filename: (None) Separator: , Decimal: . Header
Partition 70/15/15 Seed: 42 View Edit
Input Ignore Weight Calculator:
Target Data Type
Auto Categoric Numeric Survival
Welcome to Rattle (rattle.togaware.com).
Rattle is a free graphical user interface for Data Science, developed using R. R is a free software environment for statistical computing, graphics, machine learning and artificial intelligence. Together Rattle and R provide a sophisticated environment for data science, statistical analyses, and data visualisation.
See the Help menu for extensive support in using Rattle. The two books Data Mining with Rattle and R (https://bit.ly/rattle_data_mining) and The Essentials of Data Science (https://bit.ly/essentials_data_science) are available from Amazon. The Togaware Desktop Data Mining Survival Guide includes Rattle documentation and is available from datamining.togaware.com
Rattle is licensed under the GNU General Public License, Version 2. Rattle comes with ABSOLUTELY NO WARRANTY. See Help -> About for details.
Rattle Version 5.3.0. Copyright 2006-2019 Togaware Pty Ltd. Rattle is a registered trademark of Togaware Pty Ltd.
Rattle was created and implemented by Graham Williams with contributions as acknowledged in 'library(help=rattle)'.
To Begin: Choose the data source, specify the details, then click the Execute button.

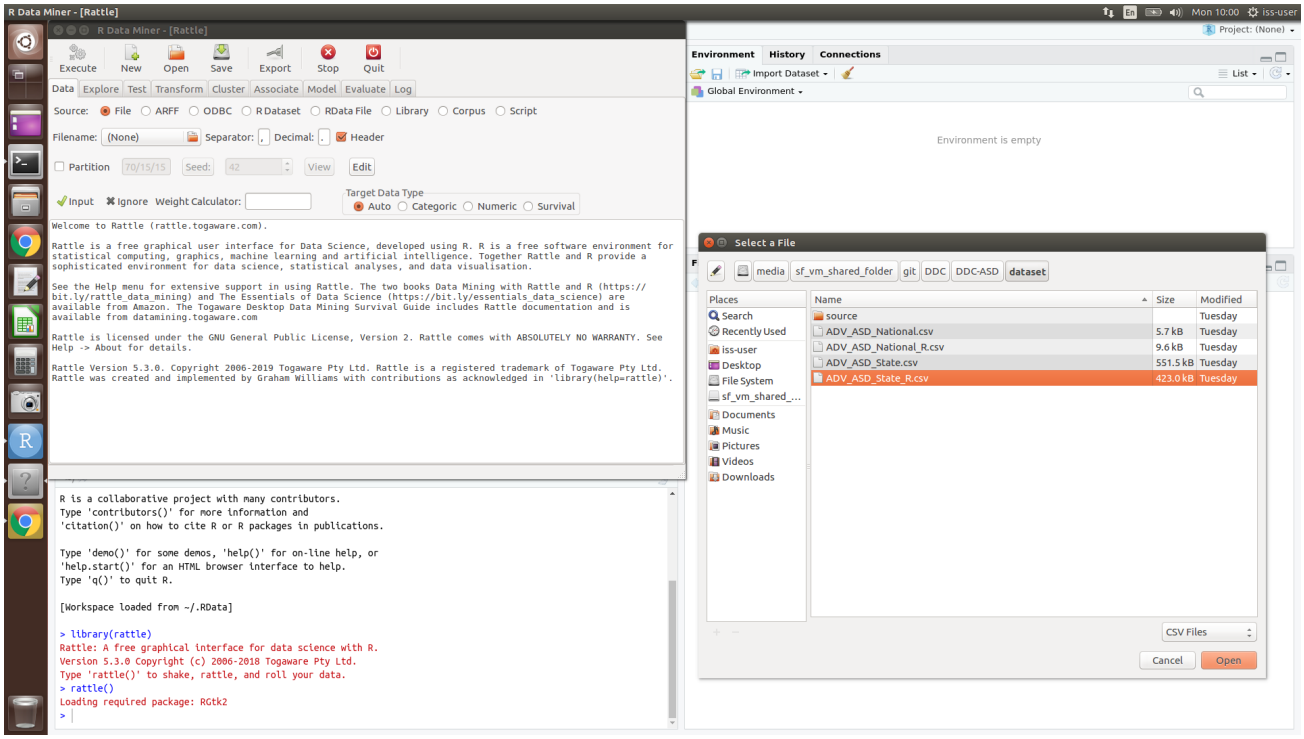
In [6]:
```
#============================================================================

# Rattle is Copyright (c) 2006-2018 Togaware Pty Ltd.
# It is free (as in libre) open source software.
# It is licensed under the GNU General Public License,
# Version 2. Rattle comes with ABSOLUTELY NO WARRANTY.
# Rattle was written by Graham Williams with contributions
# from others as acknowledged in 'library(help=rattle)'.
# Visit https://rattle.togaware.com/ for details.

#============================================================================
# Rattle timestamp: 2019-12-23 09:42:23 x86_64-pc-linux-gnu

# Rattle version 5.3.0 user 'iss-user'

# This log captures interactions with Rattle as an R script.

# For repeatability, export this activity log to a
# file, like 'model.R' using the Export button or
# through the Tools menu. Th script can then serve as a
# starting point for developing your own scripts.
# After xporting to a file called 'model.R', for exmample,
# you can type into a new R Console the command
# "source('model.R')" and so repeat all actions. Generally,
# you will want to edit the file to suit your own needs.
# You can also edit this log in place to record additional
# information before exporting the script.

# Note that saving/loading projects retains this log.

# We begin most scripts by loading the required packages.
# Here are some initial packages to load and others will be
# identified as we proceed through the script. When writing
# our own scripts we often collect together the library
# commands at the beginning of the script here.

library(rattle)   # Access the weather dataset and utilities.
library(magrittr) # Utilise %>% and %<>% pipeline operators.

# This log generally records the process of building a model.
# However, with very little effort the log can also be used
# to score a new dataset. The logical variable 'building'
# is used to toggle between generating transformations,
# when building a model and using the transformations,
# when scoring a dataset.

building <- TRUE
scoring  <- ! building

# A pre-defined value is used to reset the random seed
# so that results are repeatable.

crv$seed <- 88
set.seed(88)
```

()

Rattle: Import Data

Rattle: Import Data

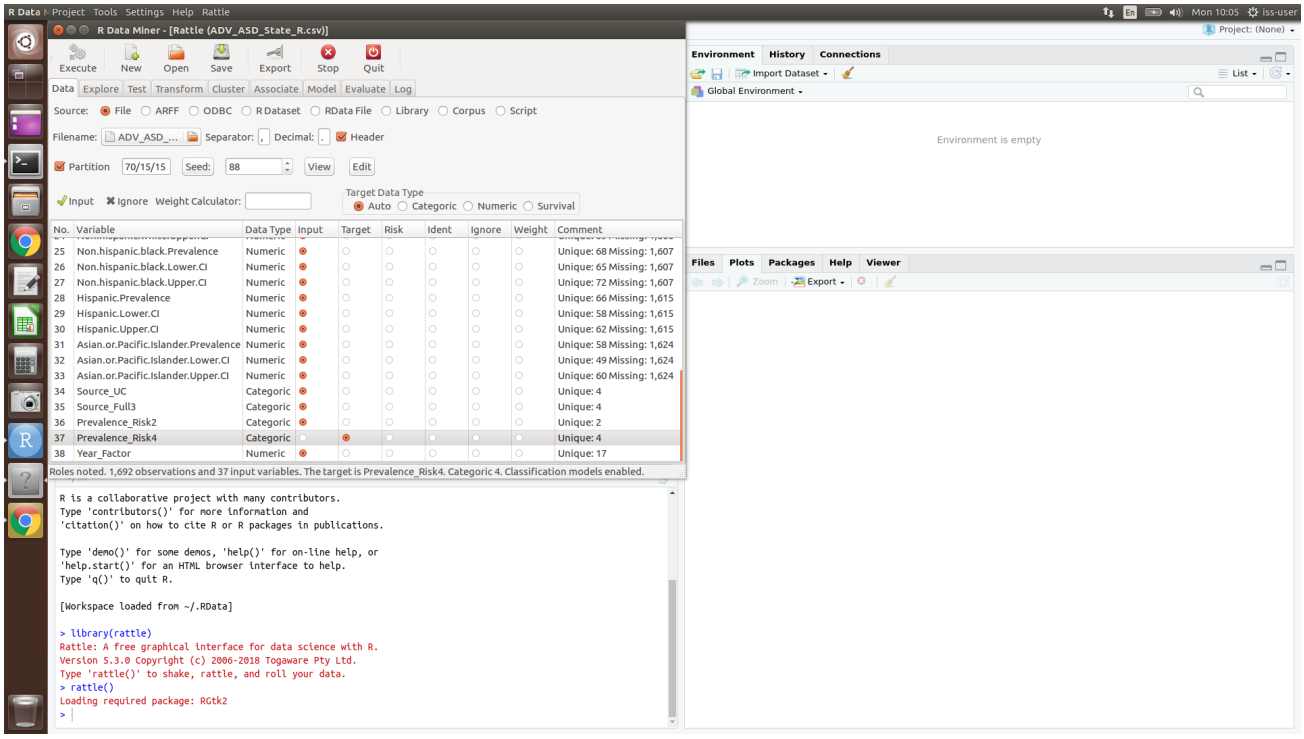
Use Case Data: "../dataset/ADV_ASD_State_R.csv"



In [7]:

```
#=======================================================================
# Rattle timestamp: 2019-12-23 10:01:39 x86_64-pc-linux-gnu

# Load a dataset from file.

fname <- "../dataset/ADV_ASD_State_R.csv"
crs$dataset <- read.csv(fname, na.strings=c(".", "NA", "", "?"), strip.white=T

#=======================================================================
# Rattle timestamp: 2019-12-23 10:01:40 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# Build the train/validate/test datasets.

# nobs=1692 train=1184 validate=254 test=254

set.seed(crv$seed)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test

# The following variable selections have been noted.

crs$input     <- c("State", "Denominator", "Prevalence",
                   "Lower.CI", "Upper.CI", "Year", "Source_Full1",
                   "State_Full1", "State_Full2", "Numerator_ASD",
                   "Numerator_NonASD", "Proportion",
                   "Chi_Wilson_Corrected_Lower.CI",
                   "Chi_Wilson_Corrected_Upper.CI",
                   "Male.Prevalence", "Male.Lower.CI",
                   "Male.Upper.CI", "Female.Prevalence",
                   "Female.Lower.CI", "Female.Upper.CI",
                   "Non.hispanic.white.Prevalence",
                   "Non.hispanic.white.Lower.CI",
                   "Non.hispanic.white.Upper.CI",
                   "Non.hispanic.black.Prevalence",
                   "Non.hispanic.black.Lower.CI",
                   "Non.hispanic.black.Upper.CI",
                   "Hispanic.Prevalence", "Hispanic.Lower.CI",
                   "Hispanic.Upper.CI",
                   "Asian.or.Pacific.Islander.Prevalence",
                   "Asian.or.Pacific.Islander.Lower.CI",
                   "Asian.or.Pacific.Islander.Upper.CI", "Source_UC",
                   "Source_Full3", "Prevalence_Risk2",
                   "Prevalence_Risk4", "Year_Factor")

crs$numeric   <- c("Denominator", "Prevalence", "Lower.CI",
                   "Upper.CI", "Year", "Numerator_ASD",
                   "Numerator_NonASD", "Proportion",
```

```
                      "Chi_Wilson_Corrected_Lower.CI",
                      "Chi_Wilson_Corrected_Upper.CI",
                      "Male.Prevalence", "Male.Lower.CI",
                      "Male.Upper.CI", "Female.Prevalence",
                      "Female.Lower.CI", "Female.Upper.CI",
                      "Non.hispanic.white.Prevalence",
                      "Non.hispanic.white.Lower.CI",
                      "Non.hispanic.white.Upper.CI",
                      "Non.hispanic.black.Prevalence",
                      "Non.hispanic.black.Lower.CI",
                      "Non.hispanic.black.Upper.CI",
                      "Hispanic.Prevalence", "Hispanic.Lower.CI",
                      "Hispanic.Upper.CI",
                      "Asian.or.Pacific.Islander.Prevalence",
                      "Asian.or.Pacific.Islander.Lower.CI",
                      "Asian.or.Pacific.Islander.Upper.CI",
                      "Year_Factor")

crs$categoric <- c("State", "Source_Full1", "State_Full1",
                   "State_Full2", "Source_UC", "Source_Full3",
                   "Prevalence_Risk2", "Prevalence_Risk4")

crs$target    <- "Source"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
```



In [8]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 10:05:28 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# Build the train/validate/test datasets.

# nobs=1692 train=1184 validate=254 test=254

set.seed(88)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test

# The following variable selections have been noted.

crs$input     <- c("State", "Denominator", "Prevalence",
                   "Lower.CI", "Upper.CI", "Year", "Source",
                   "Source_Full1", "State_Full1", "State_Full2",
                   "Numerator_ASD", "Numerator_NonASD", "Proportion",
                   "Chi_Wilson_Corrected_Lower.CI",
                   "Chi_Wilson_Corrected_Upper.CI",
                   "Male.Prevalence", "Male.Lower.CI",
                   "Male.Upper.CI", "Female.Prevalence",
                   "Female.Lower.CI", "Female.Upper.CI",
                   "Non.hispanic.white.Prevalence",
                   "Non.hispanic.white.Lower.CI",
                   "Non.hispanic.white.Upper.CI",
                   "Non.hispanic.black.Prevalence",
                   "Non.hispanic.black.Lower.CI",
                   "Non.hispanic.black.Upper.CI",
                   "Hispanic.Prevalence", "Hispanic.Lower.CI",
                   "Hispanic.Upper.CI",
                   "Asian.or.Pacific.Islander.Prevalence",
                   "Asian.or.Pacific.Islander.Lower.CI",
                   "Asian.or.Pacific.Islander.Upper.CI", "Source_UC",
                   "Source_Full3", "Prevalence_Risk2", "Year_Factor")

crs$numeric   <- c("Denominator", "Prevalence", "Lower.CI",
                   "Upper.CI", "Year", "Numerator_ASD",
                   "Numerator_NonASD", "Proportion",
                   "Chi_Wilson_Corrected_Lower.CI",
                   "Chi_Wilson_Corrected_Upper.CI",
                   "Male.Prevalence", "Male.Lower.CI",
                   "Male.Upper.CI", "Female.Prevalence",
                   "Female.Lower.CI", "Female.Upper.CI",
                   "Non.hispanic.white.Prevalence",
                   "Non.hispanic.white.Lower.CI",
                   "Non.hispanic.white.Upper.CI",
                   "Non.hispanic.black.Prevalence",
```

```
                    "Non.hispanic.black.Lower.CI",
                    "Non.hispanic.black.Upper.CI",
                    "Hispanic.Prevalence", "Hispanic.Lower.CI",
                    "Hispanic.Upper.CI",
                    "Asian.or.Pacific.Islander.Prevalence",
                    "Asian.or.Pacific.Islander.Lower.CI",
                    "Asian.or.Pacific.Islander.Upper.CI",
                    "Year_Factor")

crs$categoric <- c("State", "Source", "Source_Full1",
                    "State_Full1", "State_Full2", "Source_UC",
                    "Source_Full3", "Prevalence_Risk2")

crs$target    <- "Prevalence_Risk4"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- NULL
crs$weights   <- NULL
```

()

Rattle: EDA Explore & Test

In [9]:
```
if(!require(Hmisc)){install.packages("Hmisc")}
library('Hmisc')
```

```
Loading required package: Hmisc
Loading required package: lattice
Loading required package: survival
Loading required package: Formula
Loading required package: ggplot2
Registered S3 methods overwritten by 'ggplot2':
  method         from
  [.quosures     rlang
  c.quosures     rlang
  print.quosures rlang

Attaching package: ‘Hmisc’

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

    format.pval, units
```

In [10]:
```
if(!require(fBasics)){install.packages("fBasics")}
library('fBasics')
```

```
Loading required package: fBasics
Loading required package: timeDate
Loading required package: timeSeries
```

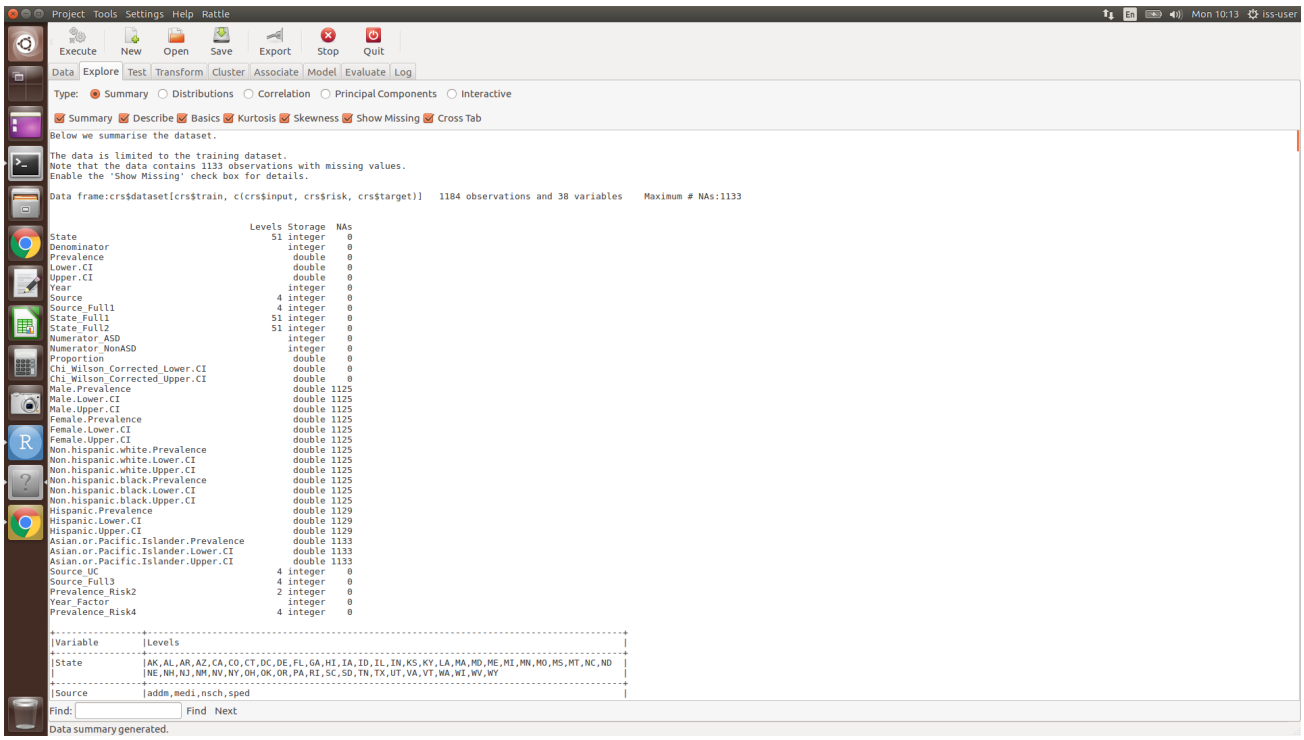
In [11]:
```
if(!require(mice)){install.packages("mice")}
library('mice')
```

```
Loading required package: mice

Attaching package: ‘mice’

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

    cbind, rbind
```

In [12]:
```
if(!require(descr)){install.packages("descr")}
library('descr')
```

```
Loading required package: descr
```

Rattle: EDA Explore & Test: Summary



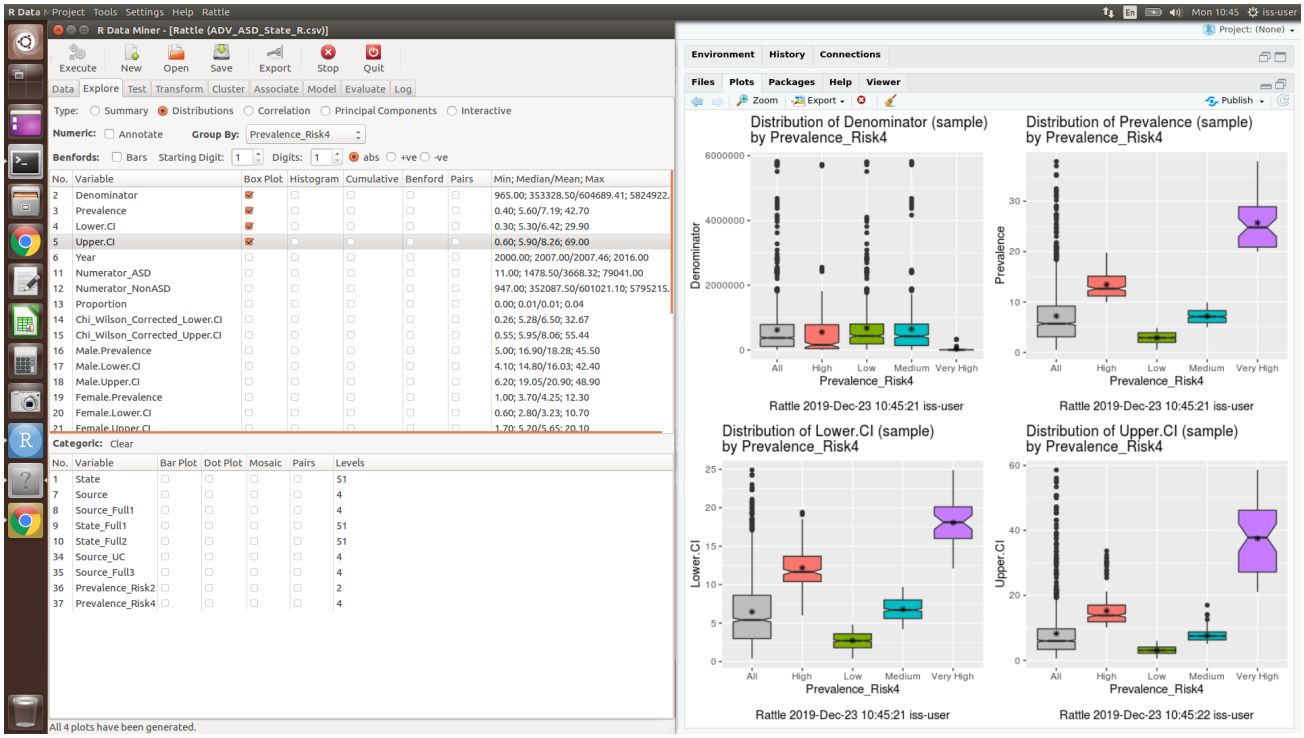
In [13]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 10:12:32 x86_64-pc-linux-gnu

# The 'Hmisc' package provides the 'contents' function.

library(Hmisc, quietly=TRUE)

# Obtain a summary of the dataset.

contents(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)])
summary(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)])

# The 'Hmisc' package provides the 'describe' function.

library(Hmisc, quietly=TRUE)

# Generate a description of the dataset.

describe(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)])

# The 'basicStats' package provides the 'fBasics' function.

library(fBasics, quietly=TRUE)

# Generate a description of the numeric data.

lapply(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)][,c(2:6, 11:3

# The 'kurtosis' package provides the 'fBasics' function.

library(fBasics, quietly=TRUE)

# Summarise the kurtosis of the numeric data.

kurtosis(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)][,c(2:6, 11

# The 'skewness' package provides the 'fBasics' function.

library(fBasics, quietly=TRUE)

# Summarise the skewness of the numeric data.

skewness(crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)][,c(2:6, 11

# The 'mice' package provides the 'md.pattern' function.

library(mice, quietly=TRUE)

# Generate a summary of the missing values in the dataset.

md.pattern(crs$dataset[,c(crs$input, crs$target)])

# The 'CrossTable' package provides the 'descr' function.

library(descr, quietly=TRUE)

# Generate cross tabulations for categoric data.

for (i in c(1, 7:10, 34:36))
{
  cat(sprintf('CrossTab of %s by target variable %s\n\n', names(crs$dataset)[i
  print(CrossTable(crs$dataset[[i]], crs$dataset[[crs$target]], expected=TRUE,
  cat(paste(rep('=', 70), collapse=''), '
```

```
')
}
```

```
Data frame:crs$dataset[crs$train, c(crs$input, crs$risk, crs$target)]   118
4 observations and 38 variables    Maximum # NAs:1133

                                        Levels Storage  NAs
State                                       51 integer    0
Denominator                                    integer    0
Prevalence                                      double    0
Lower.CI                                        double    0
Upper.CI                                        double    0
Year                                           integer    0
Source                                      4 integer    0
Source_Full1                                4 integer    0
State_Full1                                51 integer    0
State_Full2                                51 integer    0
Numerator_ASD                                  integer    0
Numerator_NonASD                               integer    0
Proportion                                      double    0
```

Rattle: EDA Explore & Test: Distribution

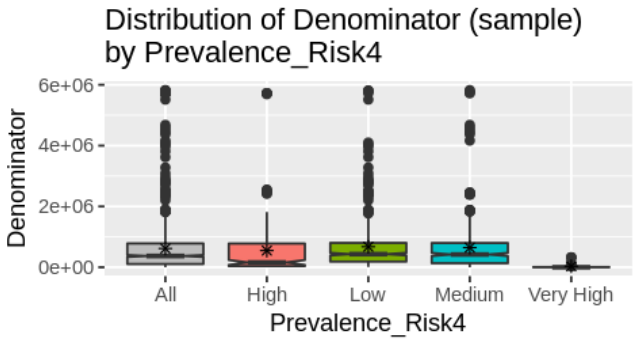


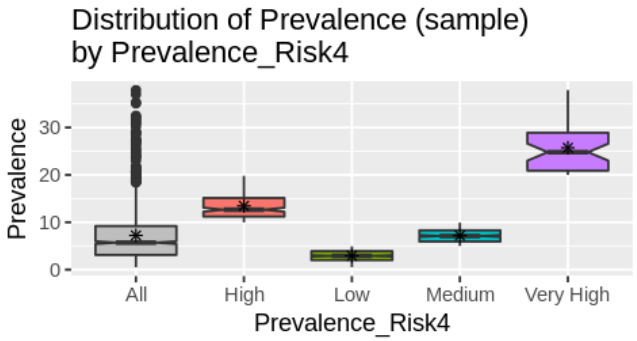
In [14]:

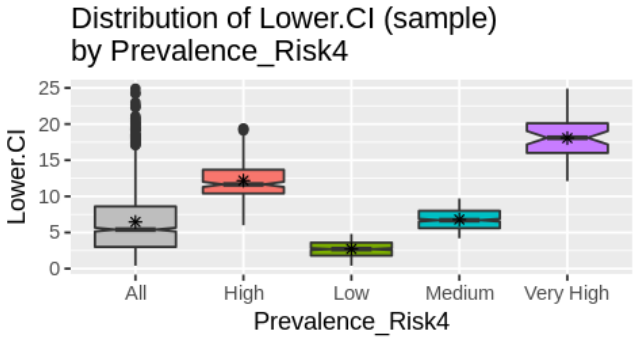
```
#=======================================================================
# Rattle timestamp: 2019-12-23 10:45:21 x86_64-pc-linux-gnu

# Display box plots for the selected variables.

# Use ggplot2 to generate box plot for Denominator

# Generate a box plot.

p01 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  ggplot2::ggplot(ggplot2::aes(y=Denominator)) +
  ggplot2::geom_boxplot(ggplot2::aes(x="All"), notch=TRUE, fill="grey") +
  ggplot2::stat_summary(ggplot2::aes(x="All"), fun.y=mean, geom="point", shape
  ggplot2::geom_boxplot(ggplot2::aes(x=Prevalence_Risk4, fill=Prevalence_Risk4
  ggplot2::stat_summary(ggplot2::aes(x=Prevalence_Risk4), fun.y=mean, geom="po
  ggplot2::xlab("Prevalence_Risk4\n\nRattle 2019-Dec-23 10:45:21 iss-user") +
  ggplot2::ggtitle("Distribution of Denominator (sample)\nby Prevalence_Risk4"
  ggplot2::theme(legend.position="none")

# Use ggplot2 to generate box plot for Prevalence

# Generate a box plot.

p02 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  ggplot2::ggplot(ggplot2::aes(y=Prevalence)) +
  ggplot2::geom_boxplot(ggplot2::aes(x="All"), notch=TRUE, fill="grey") +
  ggplot2::stat_summary(ggplot2::aes(x="All"), fun.y=mean, geom="point", shape
  ggplot2::geom_boxplot(ggplot2::aes(x=Prevalence_Risk4, fill=Prevalence_Risk4
  ggplot2::stat_summary(ggplot2::aes(x=Prevalence_Risk4), fun.y=mean, geom="po
  ggplot2::xlab("Prevalence_Risk4\n\nRattle 2019-Dec-23 10:45:21 iss-user") +
  ggplot2::ggtitle("Distribution of Prevalence (sample)\nby Prevalence_Risk4")
  ggplot2::theme(legend.position="none")

# Use ggplot2 to generate box plot for Lower.CI

# Generate a box plot.

p03 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  ggplot2::ggplot(ggplot2::aes(y=Lower.CI)) +
  ggplot2::geom_boxplot(ggplot2::aes(x="All"), notch=TRUE, fill="grey") +
  ggplot2::stat_summary(ggplot2::aes(x="All"), fun.y=mean, geom="point", shape
  ggplot2::geom_boxplot(ggplot2::aes(x=Prevalence_Risk4, fill=Prevalence_Risk4
  ggplot2::stat_summary(ggplot2::aes(x=Prevalence_Risk4), fun.y=mean, geom="po
  ggplot2::xlab("Prevalence_Risk4\n\nRattle 2019-Dec-23 10:45:21 iss-user") +
  ggplot2::ggtitle("Distribution of Lower.CI (sample)\nby Prevalence_Risk4") +
  ggplot2::theme(legend.position="none")

# Use ggplot2 to generate box plot for Upper.CI

# Generate a box plot.

p04 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  ggplot2::ggplot(ggplot2::aes(y=Upper.CI)) +
  ggplot2::geom_boxplot(ggplot2::aes(x="All"), notch=TRUE, fill="grey") +
  ggplot2::stat_summary(ggplot2::aes(x="All"), fun.y=mean, geom="point", shape
```

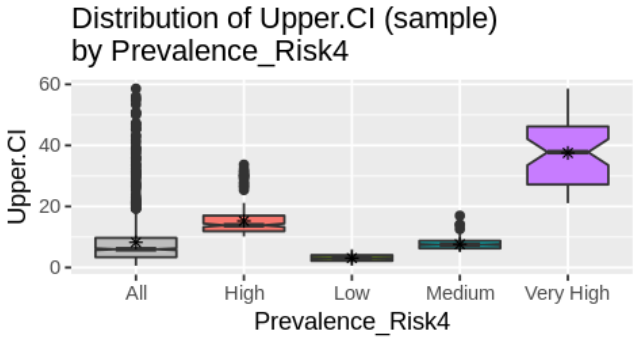
```
  ggplot2::geom_boxplot(ggplot2::aes(x=Prevalence_Risk4, fill=Prevalence_Risk4
  ggplot2::stat_summary(ggplot2::aes(x=Prevalence_Risk4), fun.y=mean, geom="po
  ggplot2::xlab("Prevalence_Risk4\n\nRattle 2019-Dec-23 10:45:22 iss-user") +
  ggplot2::ggtitle("Distribution of Upper.CI (sample)\nby Prevalence_Risk4") +
  ggplot2::theme(legend.position="none")

# Display the plots.

gridExtra::grid.arrange(p01, p02, p03, p04)
```

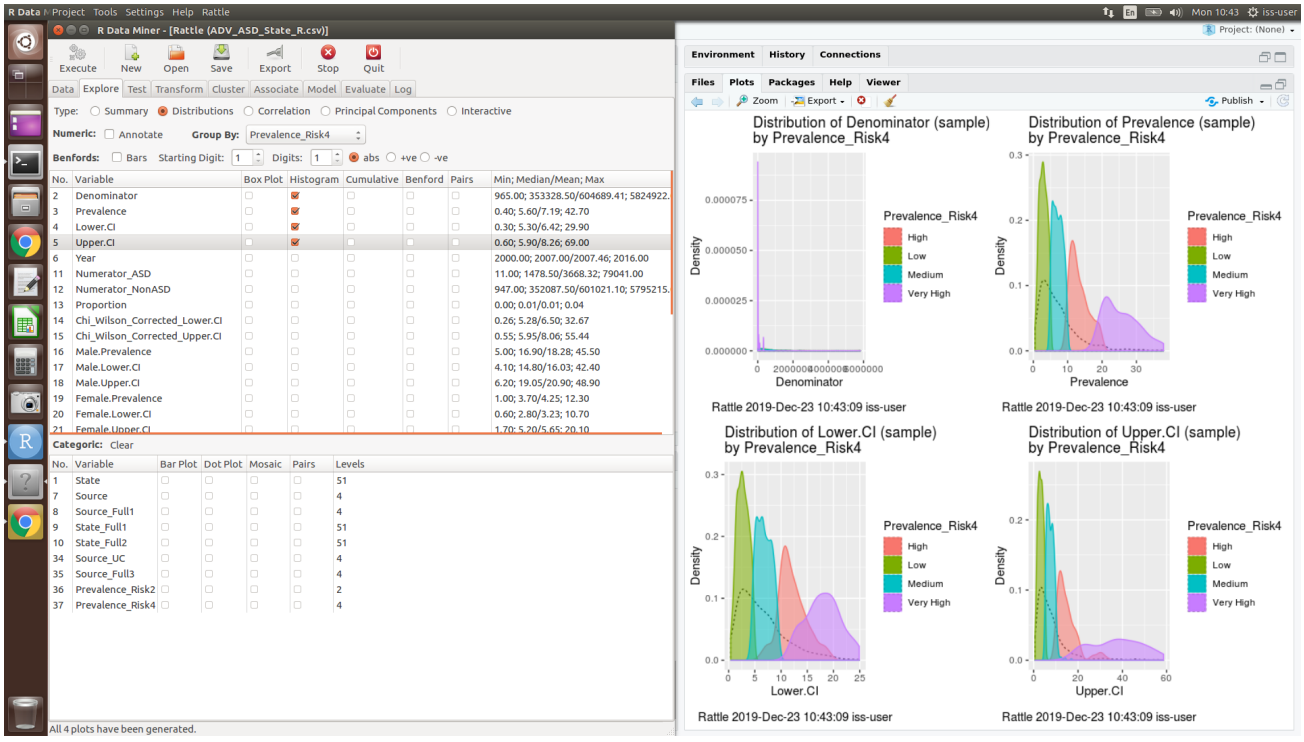
```
notch went outside hinges. Try setting notch=FALSE.
```







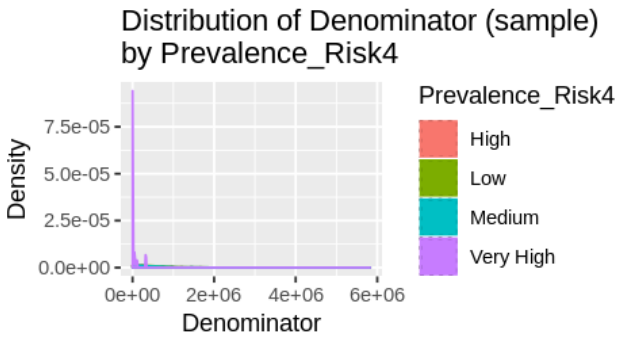




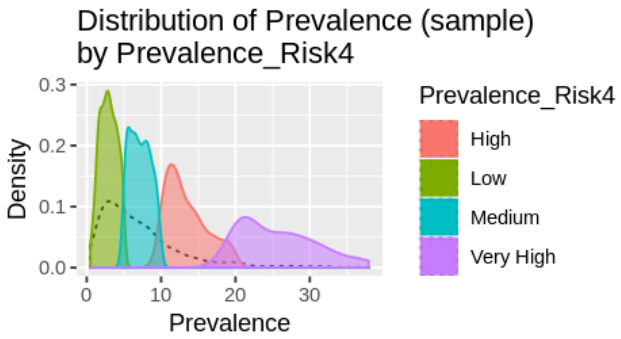
In [15]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 10:43:09 x86_64-pc-linux-gnu

# Display histogram plots for the selected variables.

# Use ggplot2 to generate histogram plot for Denominator

# Generate the plot.

p01 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Denominator, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Denominator)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Denominator\n\nRattle 2019-Dec-23 10:43:09 iss-user") +
  ggplot2::ggtitle("Distribution of Denominator (sample)\nby Prevalence_Risk4"
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Use ggplot2 to generate histogram plot for Prevalence

# Generate the plot.

p02 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Prevalence, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Prevalence)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Prevalence\n\nRattle 2019-Dec-23 10:43:09 iss-user") +
  ggplot2::ggtitle("Distribution of Prevalence (sample)\nby Prevalence_Risk4")
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Use ggplot2 to generate histogram plot for Lower.CI

# Generate the plot.

p03 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Lower.CI, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Lower.CI)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Lower.CI\n\nRattle 2019-Dec-23 10:43:09 iss-user") +
  ggplot2::ggtitle("Distribution of Lower.CI (sample)\nby Prevalence_Risk4") +
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Use ggplot2 to generate histogram plot for Upper.CI

# Generate the plot.

p04 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Upper.CI, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Upper.CI)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Upper.CI\n\nRattle 2019-Dec-23 10:43:09 iss-user") +
  ggplot2::ggtitle("Distribution of Upper.CI (sample)\nby Prevalence_Risk4") +
```

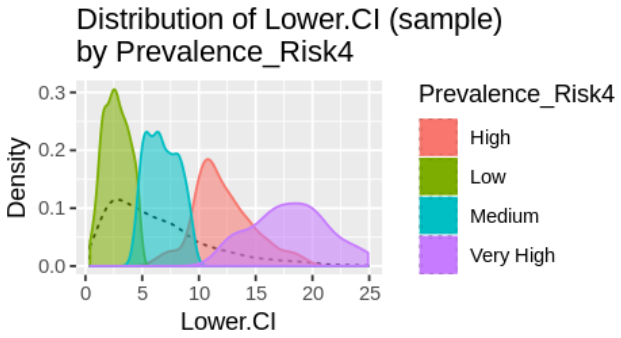
```
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Display the plots.

gridExtra::grid.arrange(p01, p02, p03, p04)
```



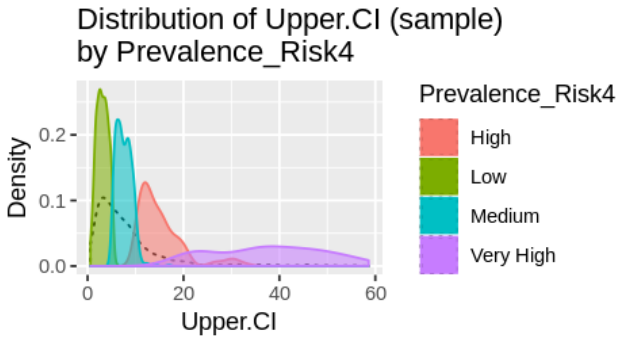
Rattle 2019-Dec-23 10:43:09 iss-user



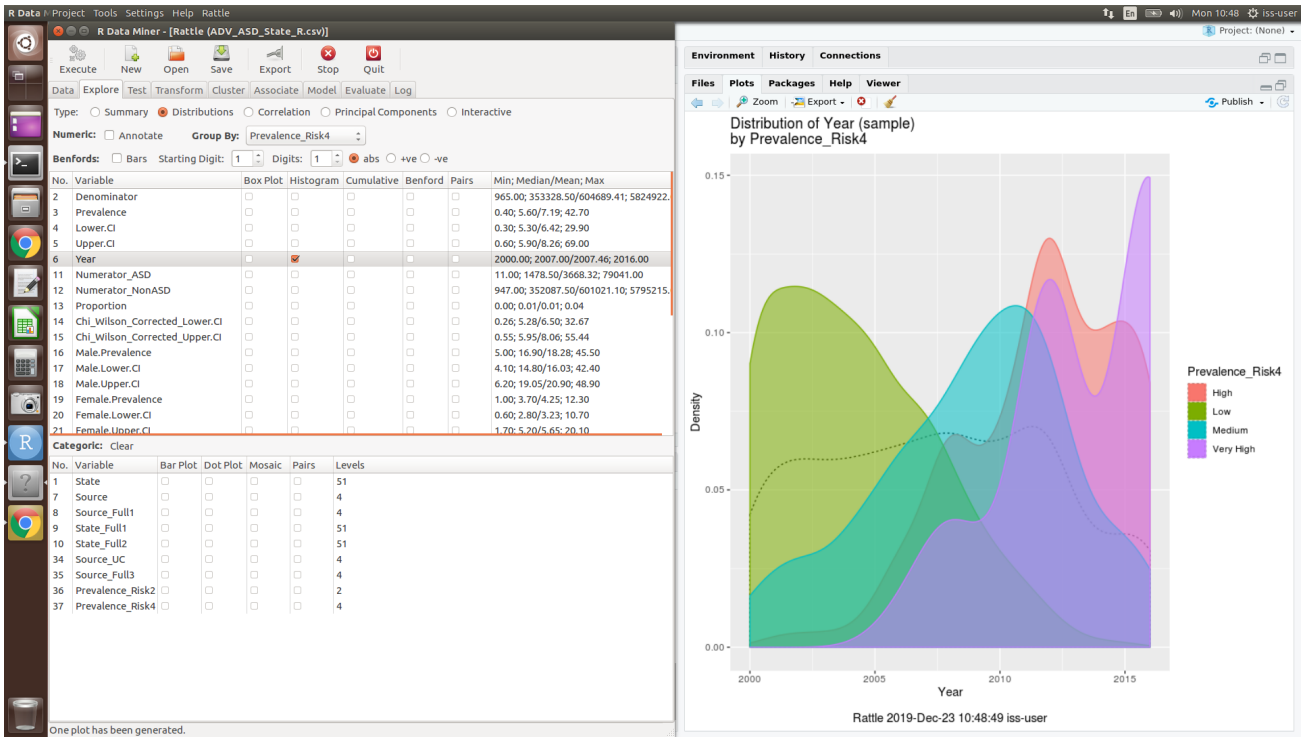
Rattle 2019-Dec-23 10:43:09 iss-user



Rattle 2019-Dec-23 10:43:09 iss-user

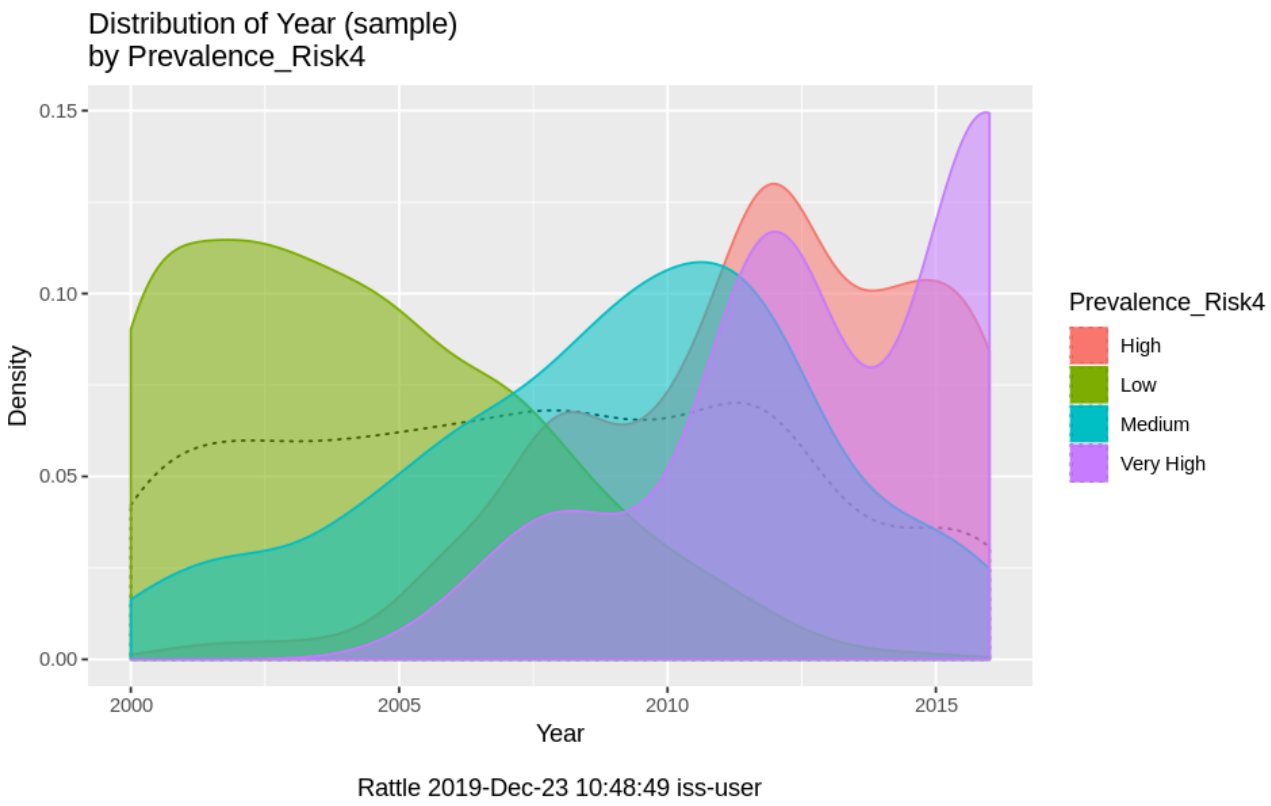


Rattle 2019-Dec-23 10:43:09 iss-user



In [16]:

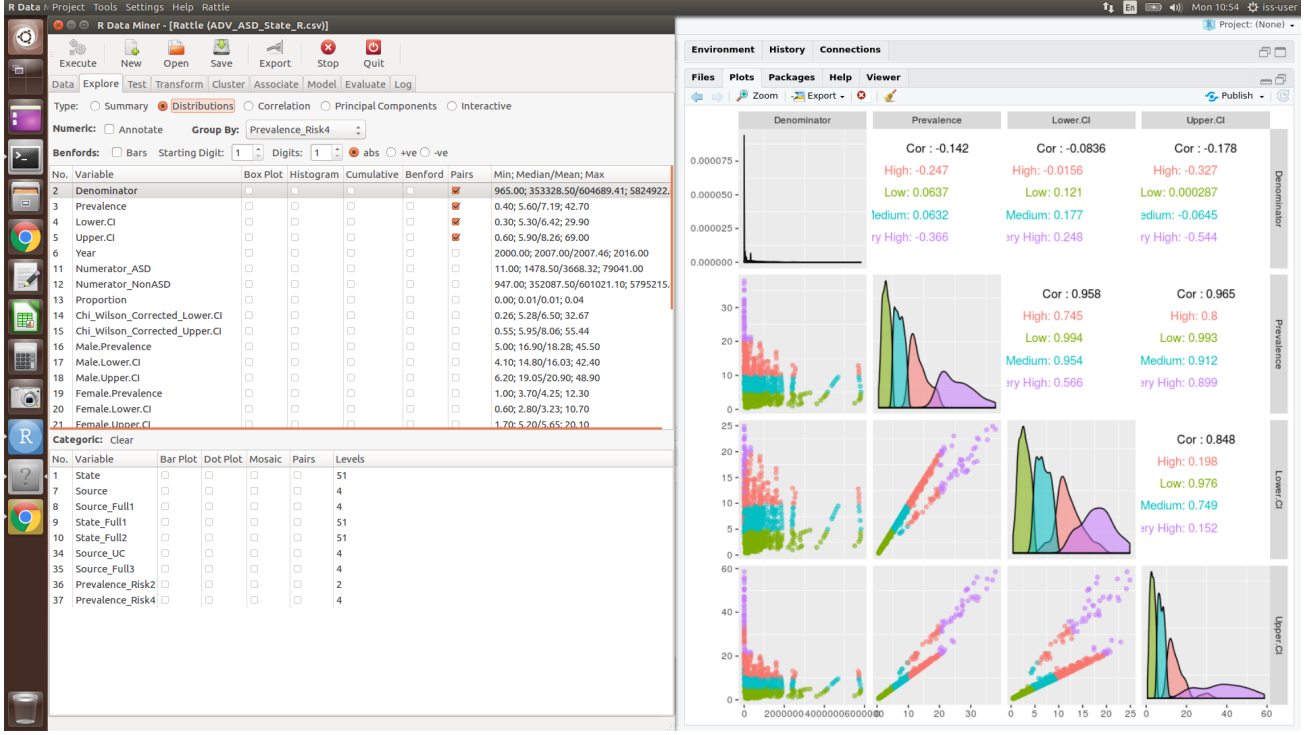
```
#============================================================================
# Rattle timestamp: 2019-12-23 10:48:49 x86_64-pc-linux-gnu

# Display histogram plots for the selected variables.

# Use ggplot2 to generate histogram plot for Year

# Generate the plot.

p01 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Year, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Year)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Year\n\nRattle 2019-Dec-23 10:48:49 iss-user") +
  ggplot2::ggtitle("Distribution of Year (sample)\nby Prevalence_Risk4") +
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Display the plots.

gridExtra::grid.arrange(p01)
```



In [17]:

```
if(!require(GGally)){install.packages("GGally")}
library('GGally')
```

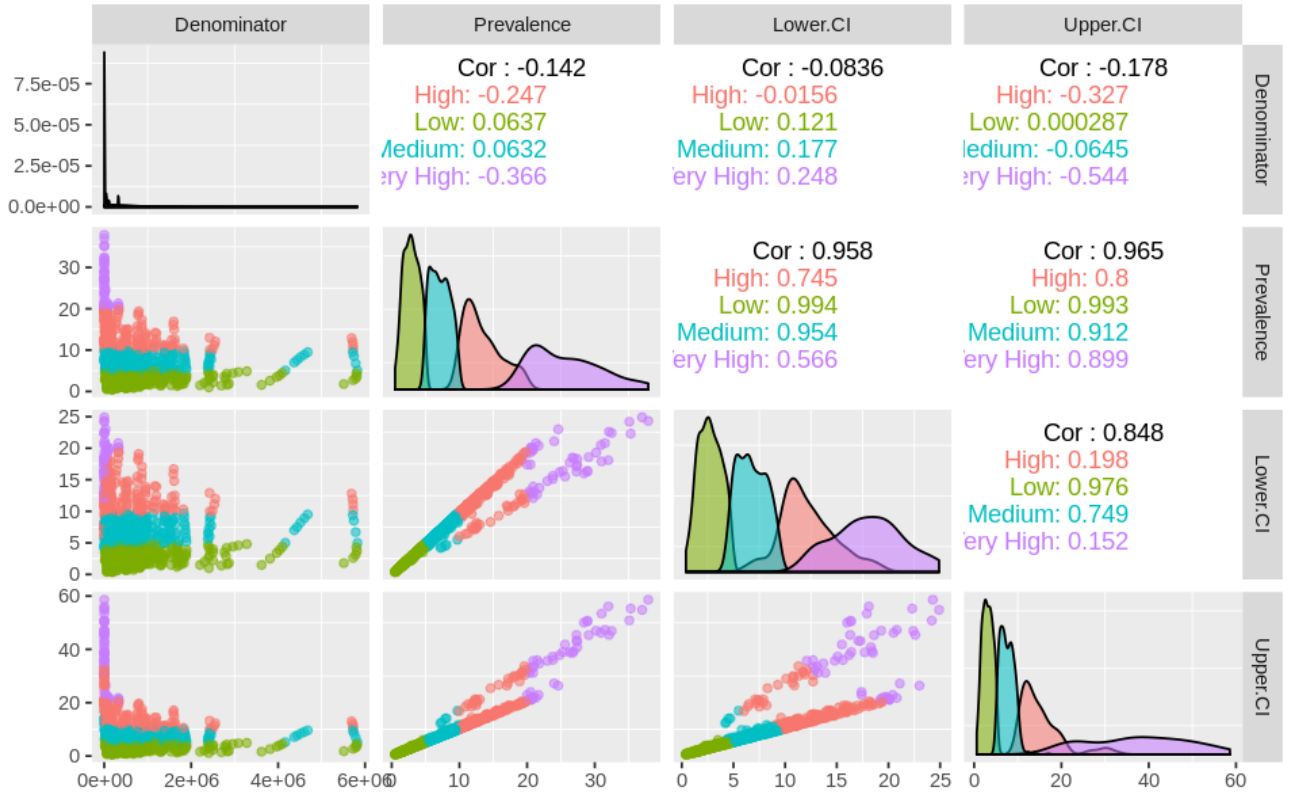
```
Loading required package: GGally
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

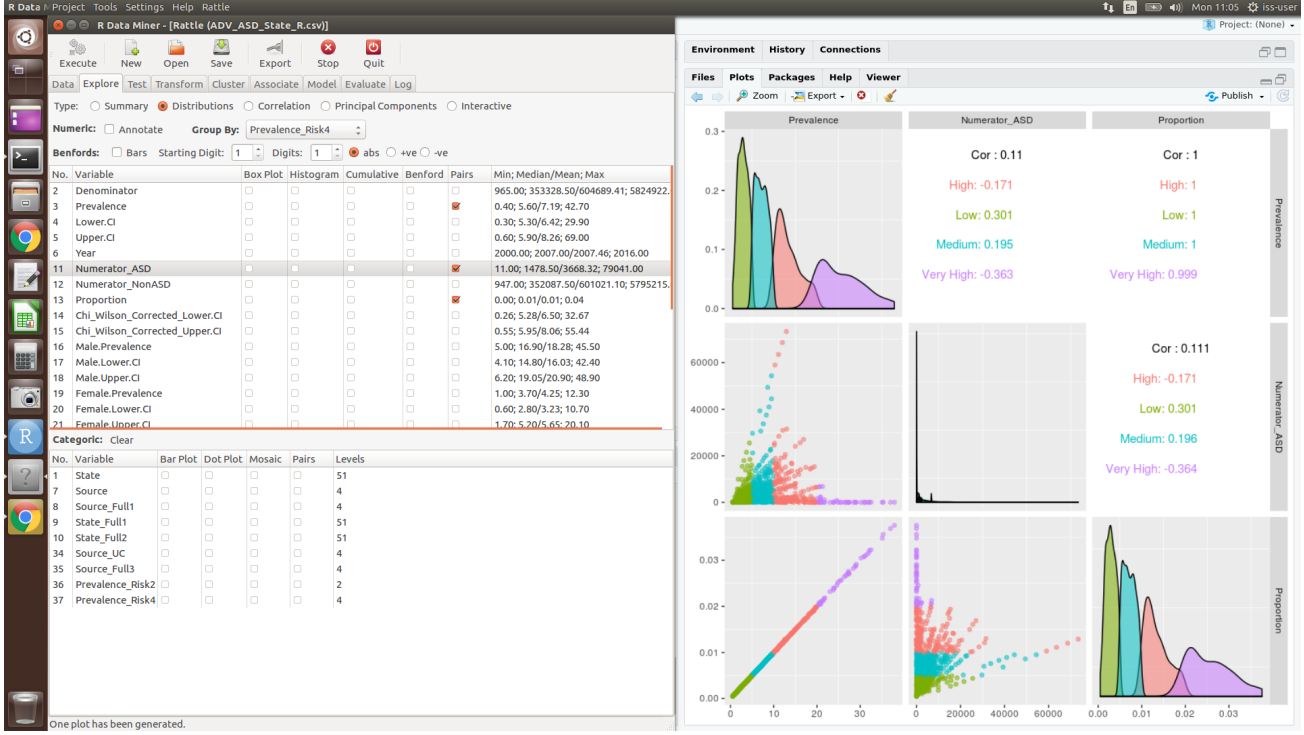

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate
Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922..
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215..
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 Source_UC 4
35 Source_Full3 4
36 Prevalence_Risk2 2
37 Prevalence_Risk4 4
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Denominator
Prevalence
Lower.CI
Upper.CI
Cor : -0.142
High: -0.247
Low: 0.0637
Cor : -0.0836
High: -0.0156
Low: 0.121
Medium: 0.177
Cor : -0.178
High: -0.327
Low: 0.000287
Cor : 0.958
High: 0.745
Low: 0.994
Medium: 0.954
Cor : 0.965
High: 0.8
Low: 0.993
Medium: 0.912
Cor : 0.848
High: 0.198
Low: 0.976
Medium: 0.749

In [18]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 10:54:13 x86_64-pc-linux-gnu

# Display a pairs plot for the selected variables.

# Use GGally's ggpairs() to do the hard work.

crs$dataset[crs$train,] %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  GGally::ggpairs(columns=c(2,3,4,5),
        mapping=ggplot2::aes(colour=Prevalence_Risk4, alpha=0.5),
                diag=list(continuous="density",
                          discrete="bar"),
                upper=list(continuous="cor",
                           combo="box",
                           discrete="ratio"),
                lower=list(continuous="points",
                           combo="denstrip",
                           discrete="facetbar")) +
  ggplot2::theme(panel.grid.major=ggplot2::element_blank())
```

```
Warning message in check_and_set_ggpairs_defaults("diag", diag, continuous =
"densityDiag", :
“Changing diag$continuous from 'density' to 'densityDiag'”Warning message in
check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", :
“Changing diag$discrete from 'bar' to 'barDiag'”
```

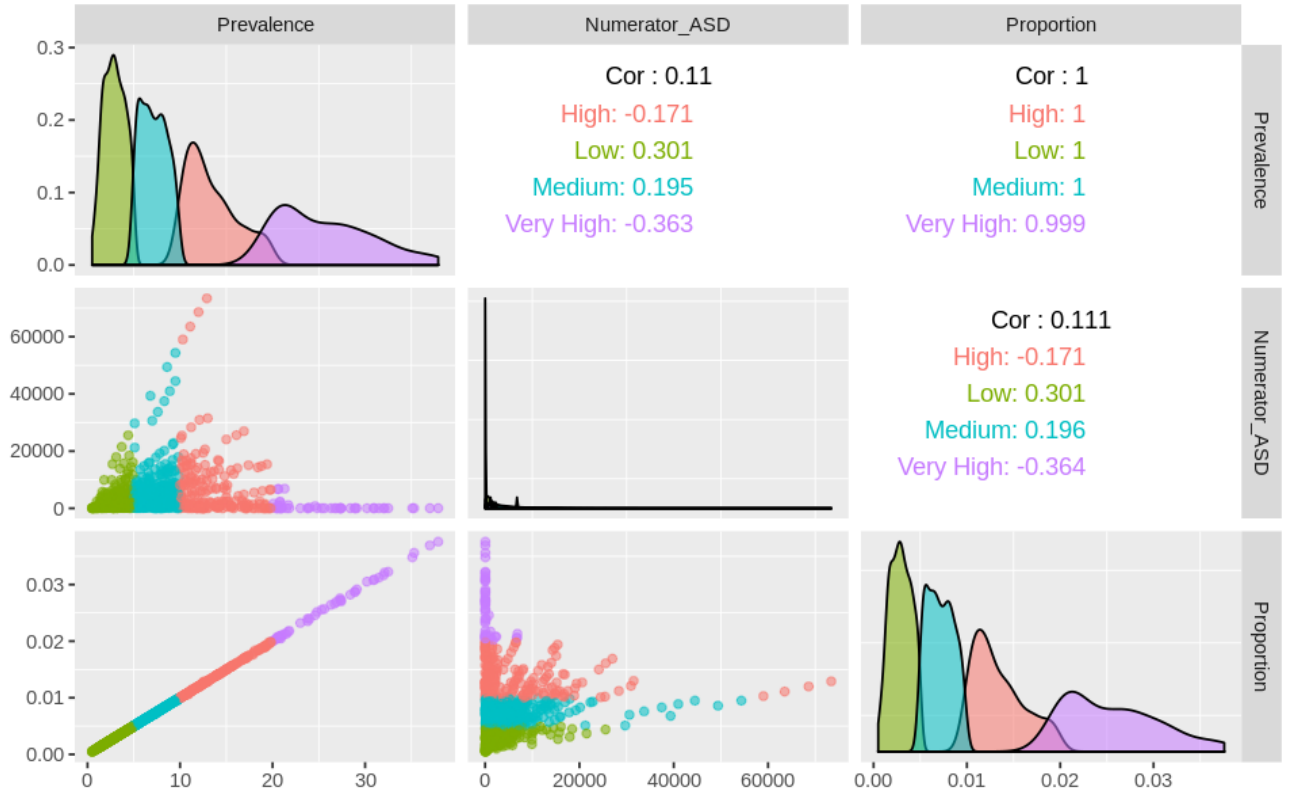


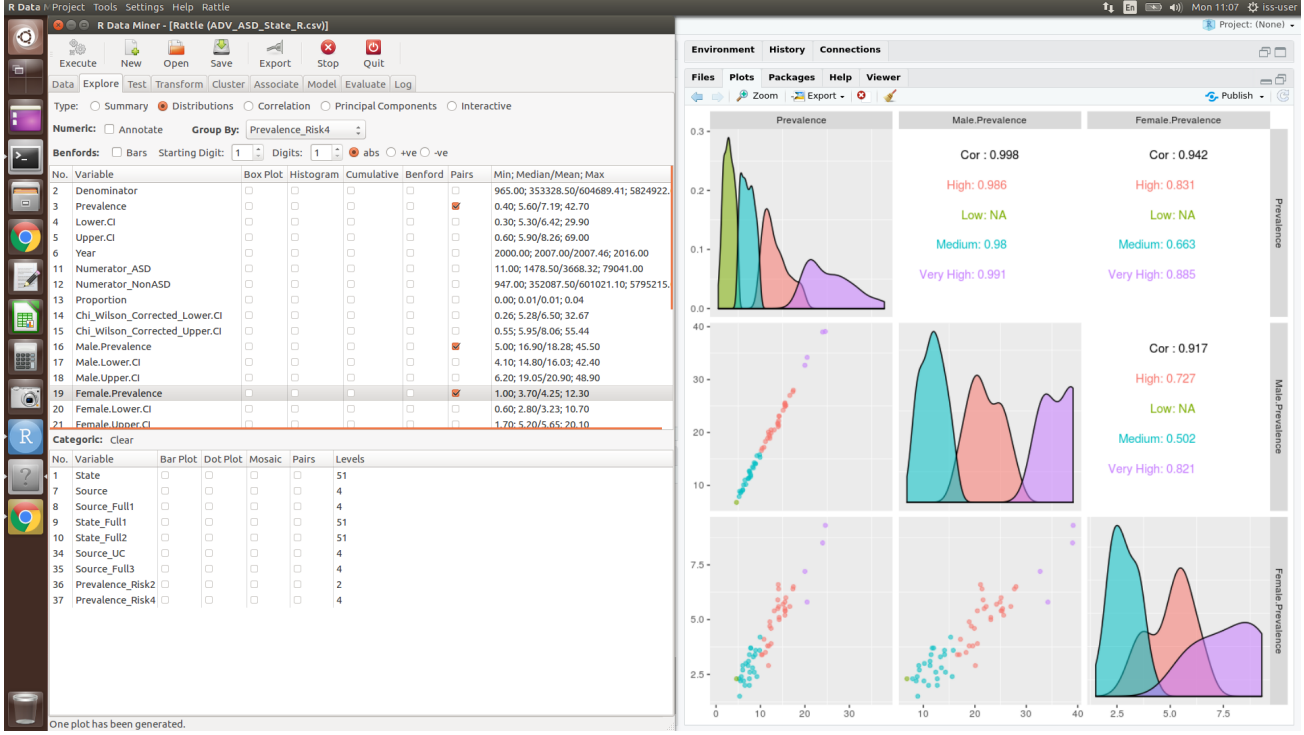

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate
Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922.
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215.
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 Source_UC 4
35 Source_Full3 4
36 Prevalence_Risk2 2
37 Prevalence_Risk4 4
One plot has been generated.
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Prevalence
Numerator_ASD
Proportion
Cor : 0.11
High: -0.171
Low: 0.301
Medium: 0.195
Very High: -0.363
Cor : 1
High: 1
Low: 1
Medium: 1
Very High: 0.999
Cor : 0.111
High: -0.171
Low: 0.301
Medium: 0.196
Very High: -0.364

In [19]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:06:10 x86_64-pc-linux-gnu

# Display a pairs plot for the selected variables.

# Use GGally's ggpairs() to do the hard work.

crs$dataset[crs$train,] %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  GGally::ggpairs(columns=c(3,11,13),
        mapping=ggplot2::aes(colour=Prevalence_Risk4, alpha=0.5),
                diag=list(continuous="density",
                          discrete="bar"),
                upper=list(continuous="cor",
                           combo="box",
                           discrete="ratio"),
                lower=list(continuous="points",
                           combo="denstrip",
                           discrete="facetbar")) +
  ggplot2::theme(panel.grid.major=ggplot2::element_blank())
```

```
Warning message in check_and_set_ggpairs_defaults("diag", diag, continuous =
"densityDiag", :
“Changing diag$continuous from 'density' to 'densityDiag'”Warning message in
check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", :
“Changing diag$discrete from 'bar' to 'barDiag'”
```

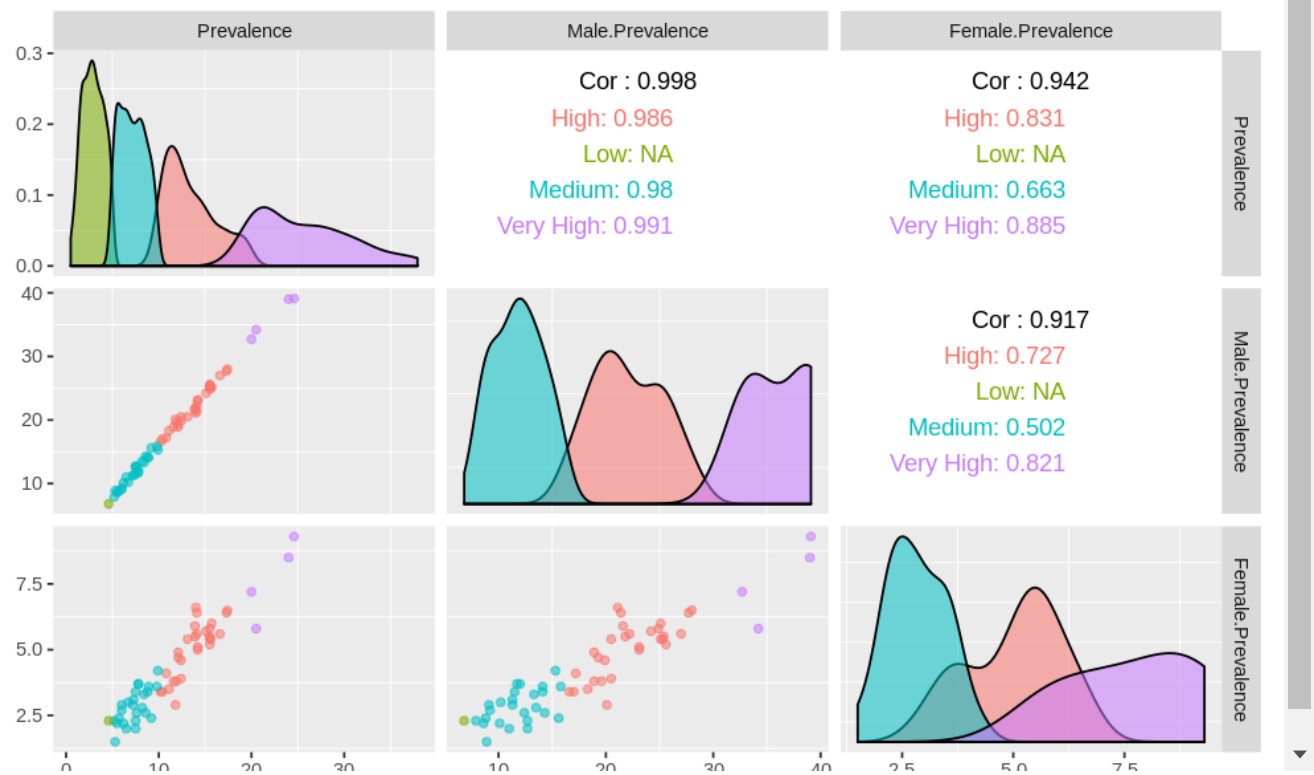


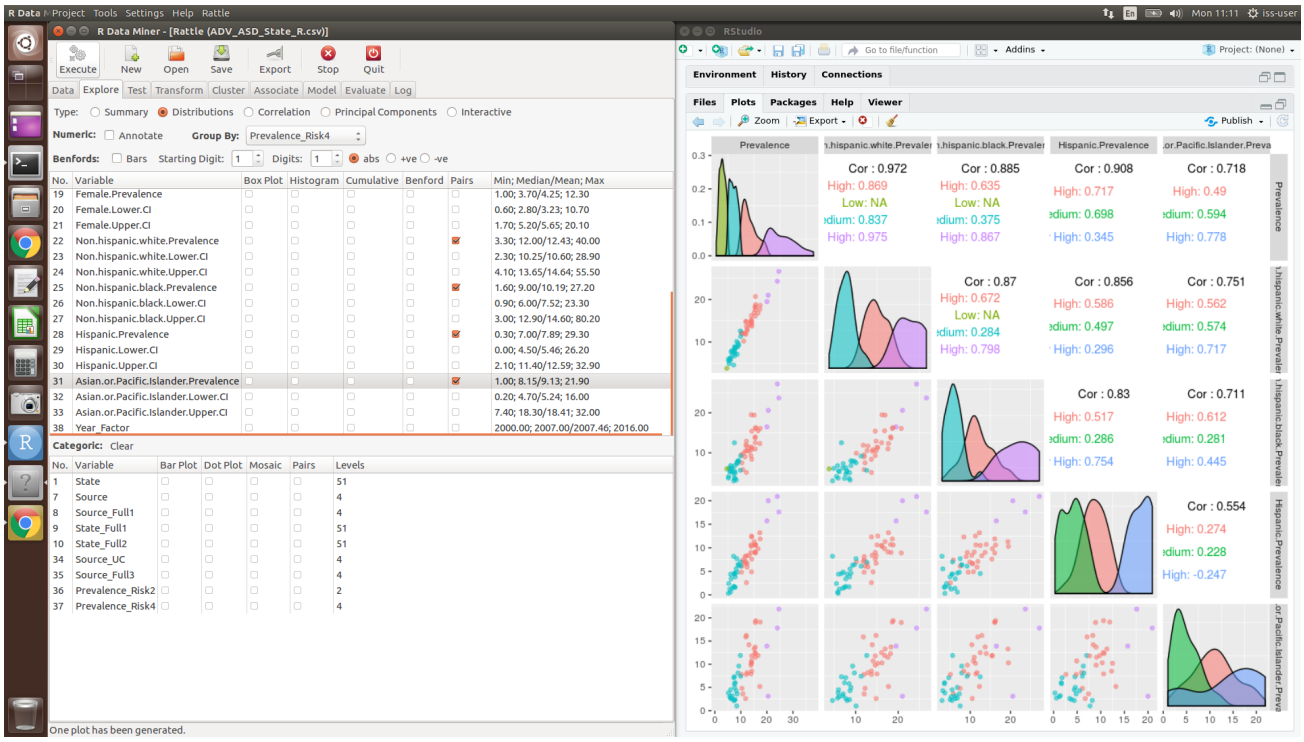

R Data Project Tools Settings Help Rattle
Mon 11:07 iss-user
R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922.
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215.
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 Source_UC 4
35 Source_Full3 4
36 Prevalence_Risk2 2
37 Prevalence_Risk4 4
One plot has been generated.
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom Export Publish
Prevalence
Male.Prevalence
Female.Prevalence
Cor : 0.998
High: 0.986
Low: NA
Medium: 0.98
Very High: 0.991
Cor : 0.942
High: 0.831
Low: NA
Medium: 0.663
Very High: 0.885
Cor : 0.917
High: 0.727
Low: NA
Medium: 0.502
Very High: 0.821
Prevalence
Male Prevalence
Female Prevalence

In [20]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:07:28 x86_64-pc-linux-gnu

# Display a pairs plot for the selected variables.

# Use GGally's ggpairs() to do the hard work.

crs$dataset[crs$train,] %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  GGally::ggpairs(columns=c(3,16,19),
        mapping=ggplot2::aes(colour=Prevalence_Risk4, alpha=0.5),
                diag=list(continuous="density",
                          discrete="bar"),
                upper=list(continuous="cor",
                           combo="box",
                           discrete="ratio"),
                lower=list(continuous="points",
                           combo="denstrip",
                           discrete="facetbar")) +
  ggplot2::theme(panel.grid.major=ggplot2::element_blank())
```

```
Warning message in check_and_set_ggpairs_defaults("diag", diag, continuous =
"densityDiag", :
“Changing diag$continuous from 'density' to 'densityDiag'”Warning message in
check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", :
“Changing diag$discrete from 'bar' to 'barDiag'”Warning message in (function
(data, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1125 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1125 rows containing missing values”Warning message:
“Removed 1125 rows containing missing values (geom_point).”Warning message:
“Removed 1125 rows containing non-finite values (stat_density).”Warning messa
ge:
“Groups with fewer than two data points have been dropped.”Warning message in
(function (data, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1125 rows containing missing values”Warning message:
“Removed 1125 rows containing missing values (geom_point).”Warning message:
“Removed 1125 rows containing missing values (geom_point).”Warning message:
“Removed 1125 rows containing non-finite values (stat_density).”Warning messa
ge:
“Groups with fewer than two data points have been dropped.”
```


Prevalence
Male.Prevalence
Female.Prevalence
Cor : 0.998
High: 0.986
Low: NA
Medium: 0.98
Very High: 0.991
Cor : 0.942
High: 0.831
Low: NA
Medium: 0.663
Very High: 0.885
Cor : 0.917
High: 0.727
Low: NA
Medium: 0.502
Very High: 0.821
Prevalence
Male.Prevalence
Female.Prevalence

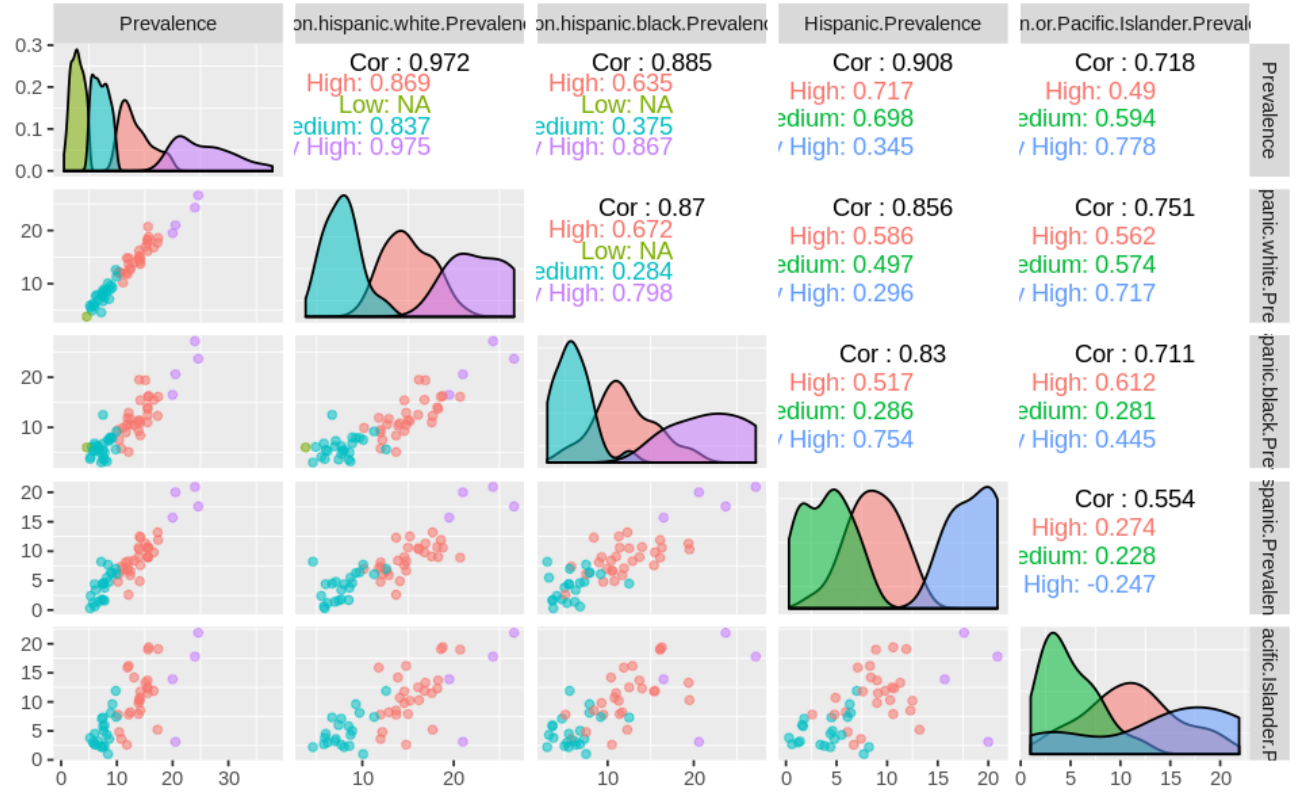

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
One plot has been generated.
RStudio
Environment
History
Connections
Files
Plots
Packages
Help
Viewer

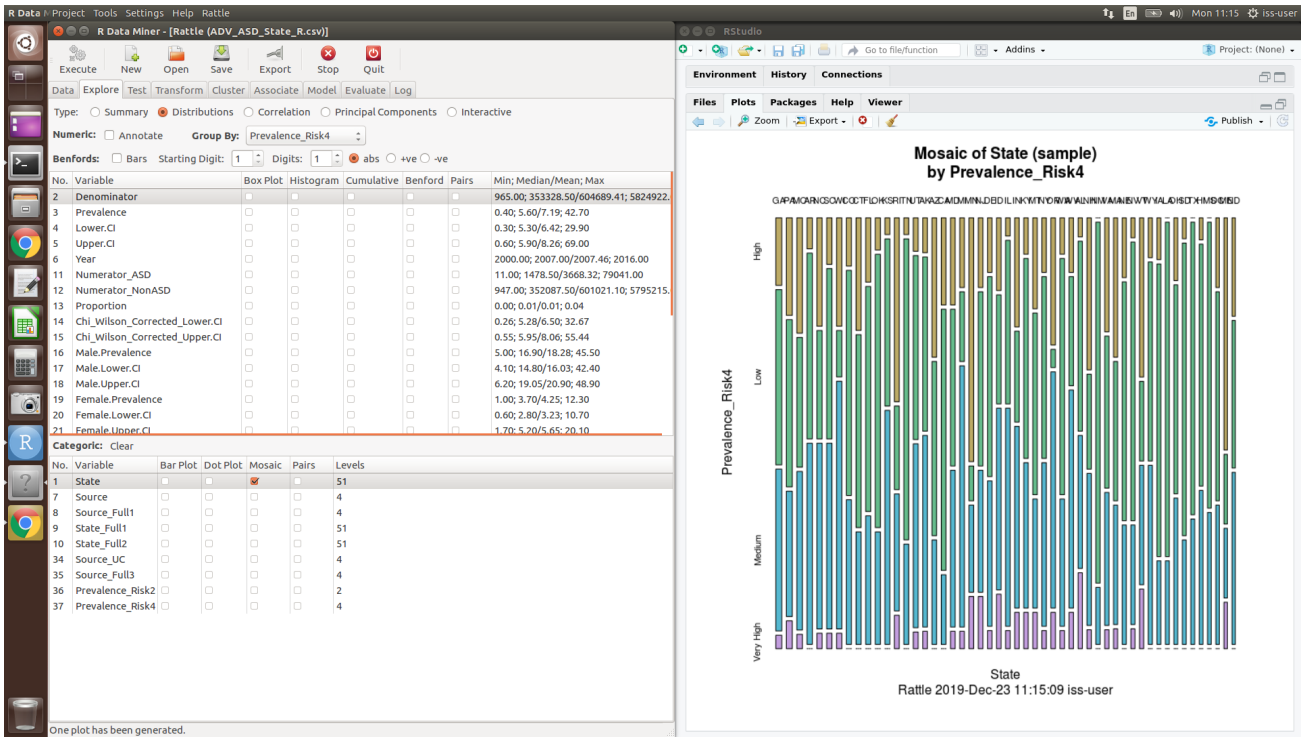
In [21]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:11:18 x86_64-pc-linux-gnu

# Display a pairs plot for the selected variables.

# Use GGally's ggpairs() to do the hard work.

crs$dataset[crs$train,] %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  GGally::ggpairs(columns=c(3,22,25,28,31),
        mapping=ggplot2::aes(colour=Prevalence_Risk4, alpha=0.5),
                diag=list(continuous="density",
                          discrete="bar"),
                upper=list(continuous="cor",
                           combo="box",
                           discrete="ratio"),
                lower=list(continuous="points",
                           combo="denstrip",
                           discrete="facetbar")) +
  ggplot2::theme(panel.grid.major=ggplot2::element_blank())
```

```
Warning message in check_and_set_ggpairs_defaults("diag", diag, continuous =
"densityDiag", :
“Changing diag$continuous from 'density' to 'densityDiag'”Warning message in
check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", :
“Changing diag$discrete from 'bar' to 'barDiag'”Warning message in (function
(data, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1125 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1125 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1129 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1133 rows containing missing values”Warning message:
“Removed 1125 rows containing missing values (geom_point).”Warning message:
“Removed 1125 rows containing non-finite values (stat_density).”Warning messa
ge:
“Groups with fewer than two data points have been dropped.”Warning message in
(function (data, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1125 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1129 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1133 rows containing missing values”Warning message:
“Removed 1125 rows containing missing values (geom_point).”Warning message:
“Removed 1125 rows containing missing values (geom_point).”Warning message:
“Removed 1125 rows containing non-finite values (stat_density).”Warning messa
ge:
“Groups with fewer than two data points have been dropped.”Warning message in
(function (data, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1129 rows containing missing values”Warning message in (function (da
ta, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1133 rows containing missing values”Warning message:
“Removed 1129 rows containing missing values (geom_point).”Warning message:
“Removed 1129 rows containing missing values (geom_point).”Warning message:
“Removed 1129 rows containing missing values (geom_point).”Warning message:
“Removed 1129 rows containing non-finite values (stat_density).”Warning messa
ge in (function (data, mapping, alignPercent = 0.6, method = "pearson", :
“Removed 1133 rows containing missing values”Warning message:
“Removed 1133 rows containing missing values (geom_point).”Warning message:
“Removed 1133 rows containing missing values (geom_point).”Warning message:
“Removed 1133 rows containing missing values (geom_point).”Warning message:
```

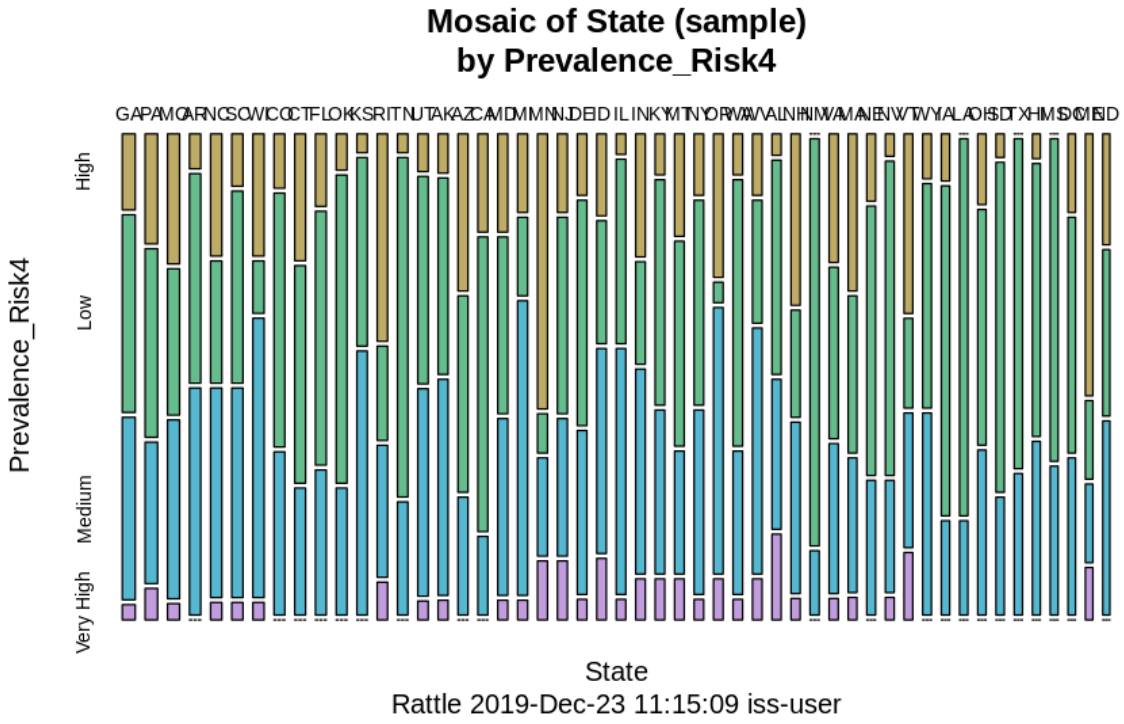
```
“Removed 1133 rows containing missing values (geom_point).”Warning message:
“Removed 1133 rows containing non-finite values (stat_density).”
```

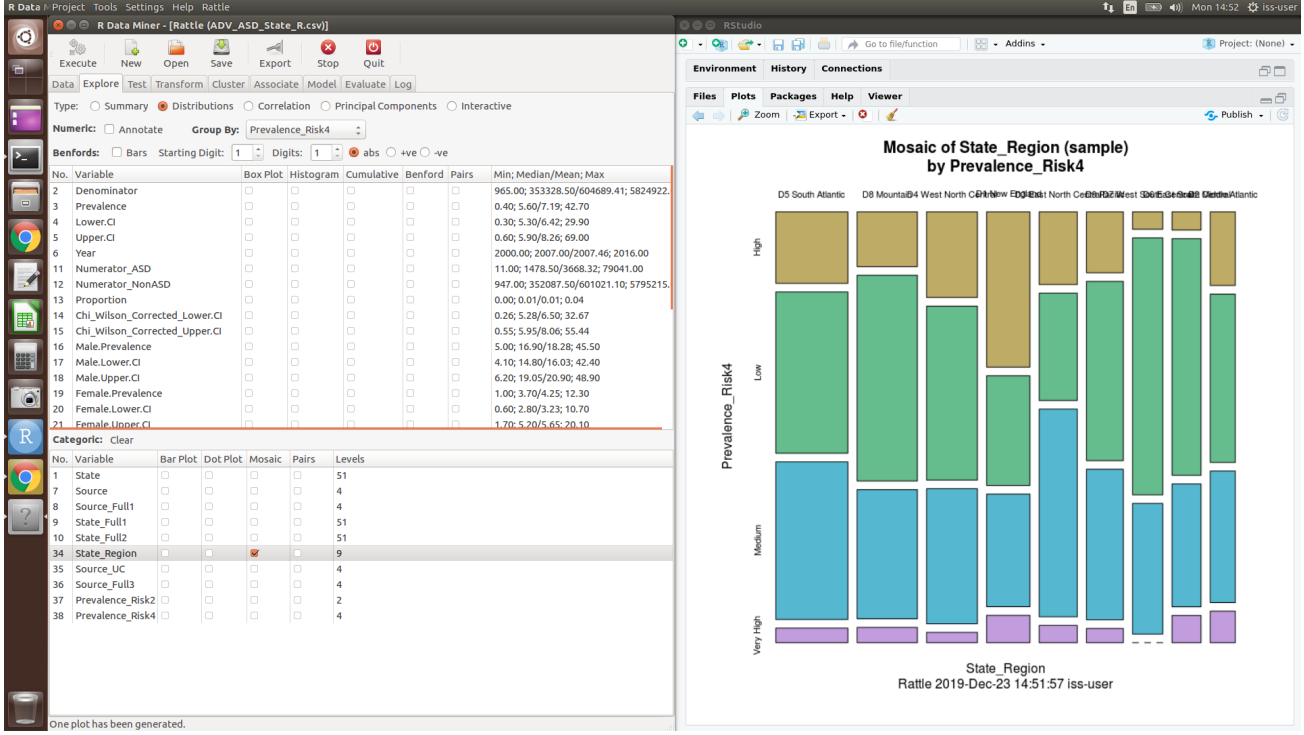




In [22]:

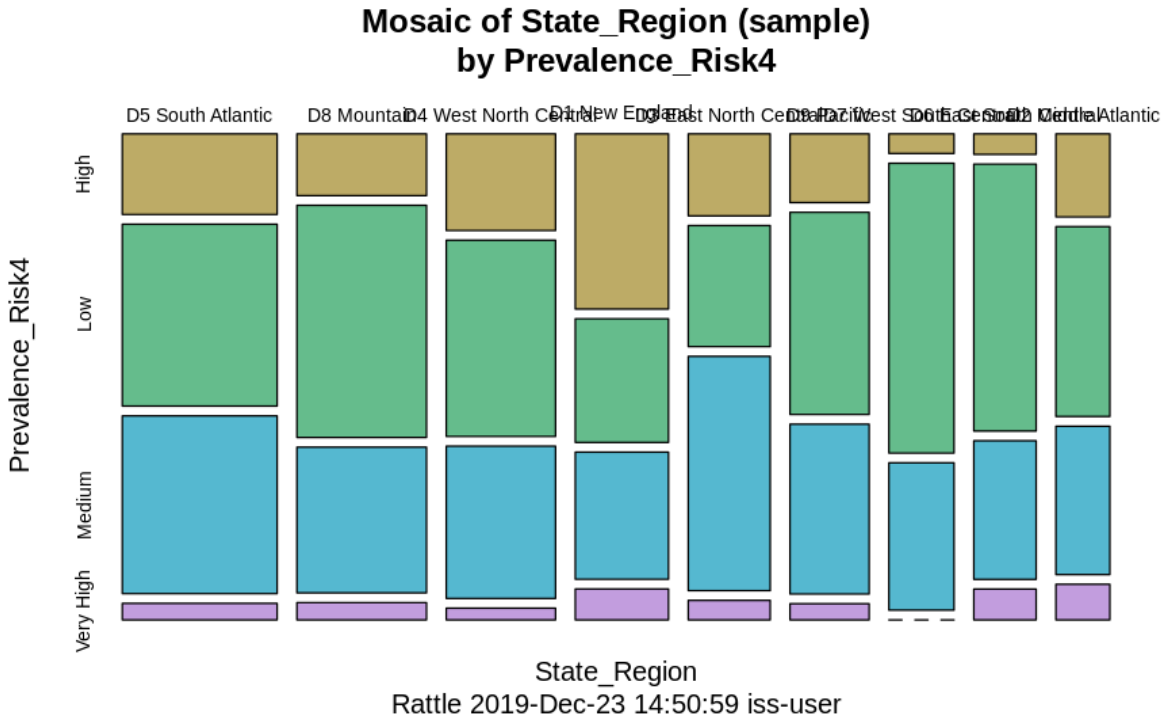
```
#============================================================================
# Rattle timestamp: 2019-12-23 11:15:09 x86_64-pc-linux-gnu

# Mosaic Plot

# Generate the table data for plotting.

ds <- table(crs$dataset[crs$train,]$State, crs$dataset[crs$train,]$Prevalence_

# Sort the entries.

ord <- order(apply(ds, 1, sum), decreasing=TRUE)

# Plot the data.

mosaicplot(ds[ord,], main="Mosaic of State (sample)
by Prevalence_Risk4", sub="Rattle 2019-Dec-23 11:15:09 iss-user", color=colors
```




R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate
Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922.
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215.
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 State_Region 9
35 Source_UC 4
36 Source_Full3 4
37 Prevalence_Risk2 2
38 Prevalence_Risk4 4
One plot has been generated.
RStudio
Addins
Project: (None)
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Mosaic of State_Region (sample)
by Prevalence_Risk4
D5 South Atlantic
D8 Mountain
Prevalence_Risk4
High
Low
Medium
Very High
State_Region
Rattle 2019-Dec-23 14:51:57 iss-user

In [23]:

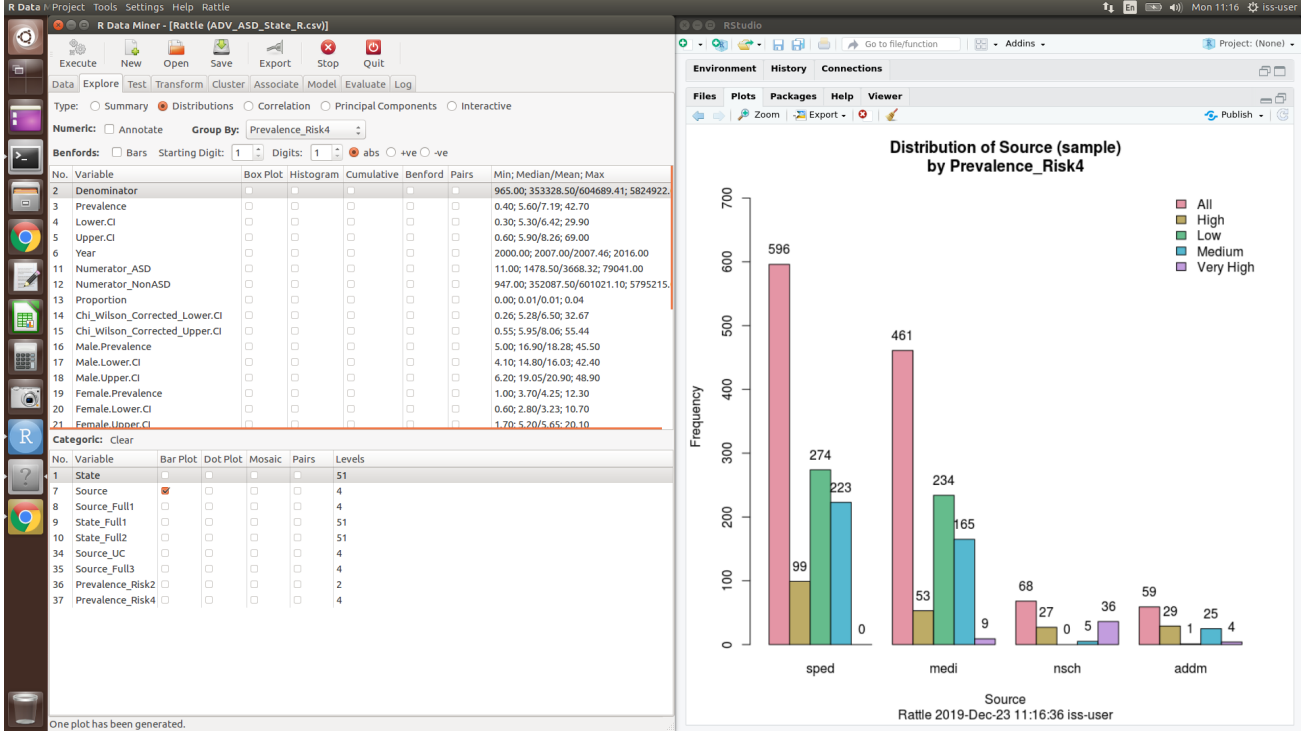
```
#=======================================================================
# Rattle timestamp: 2019-12-23 14:50:59 x86_64-pc-linux-gnu

# Mosaic Plot

# Generate the table data for plotting.

ds <- table(crs$dataset[crs$train,]$State_Region, crs$dataset[crs$train,]$Prev

# Sort the entries.

ord <- order(apply(ds, 1, sum), decreasing=TRUE)

# Plot the data.

mosaicplot(ds[ord,], main="Mosaic of State_Region (sample)
by Prevalence_Risk4", sub="Rattle 2019-Dec-23 14:50:59 iss-user", color=colors
```



In [24]:

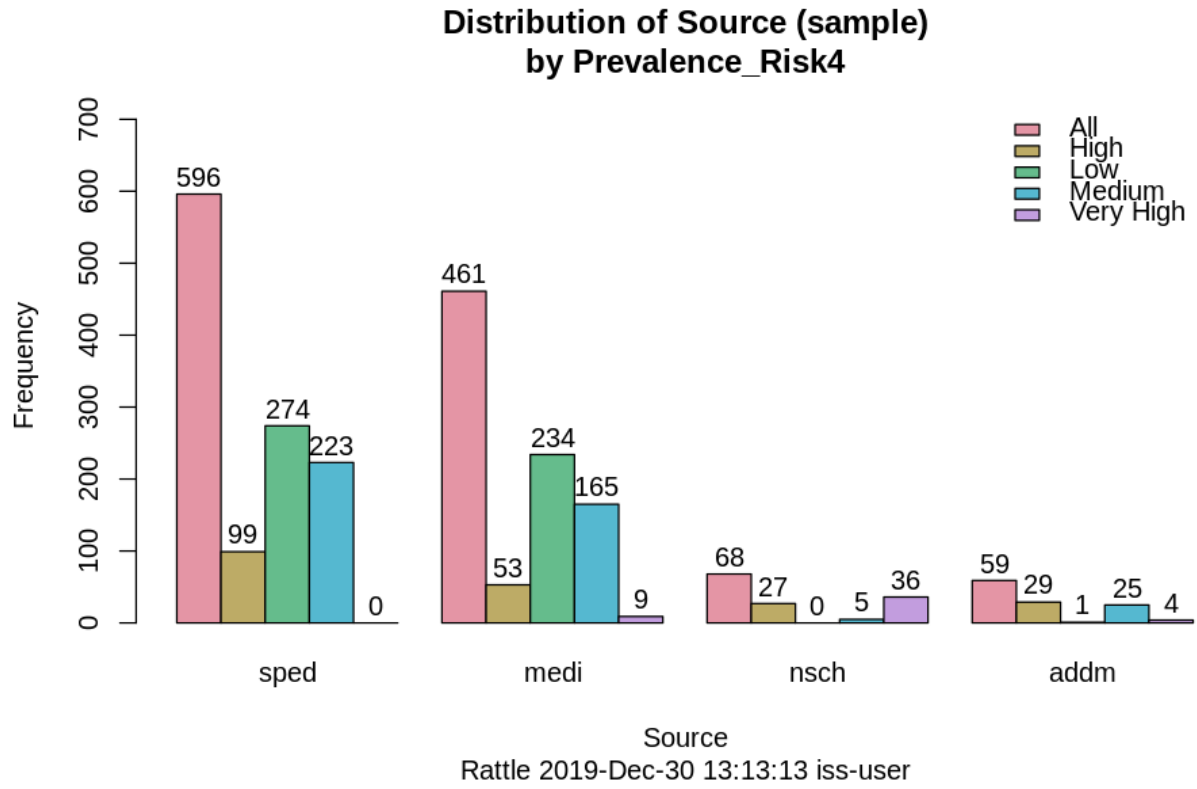
```
if(!require(gplots)){install.packages("gplots")}
library('gplots')
```

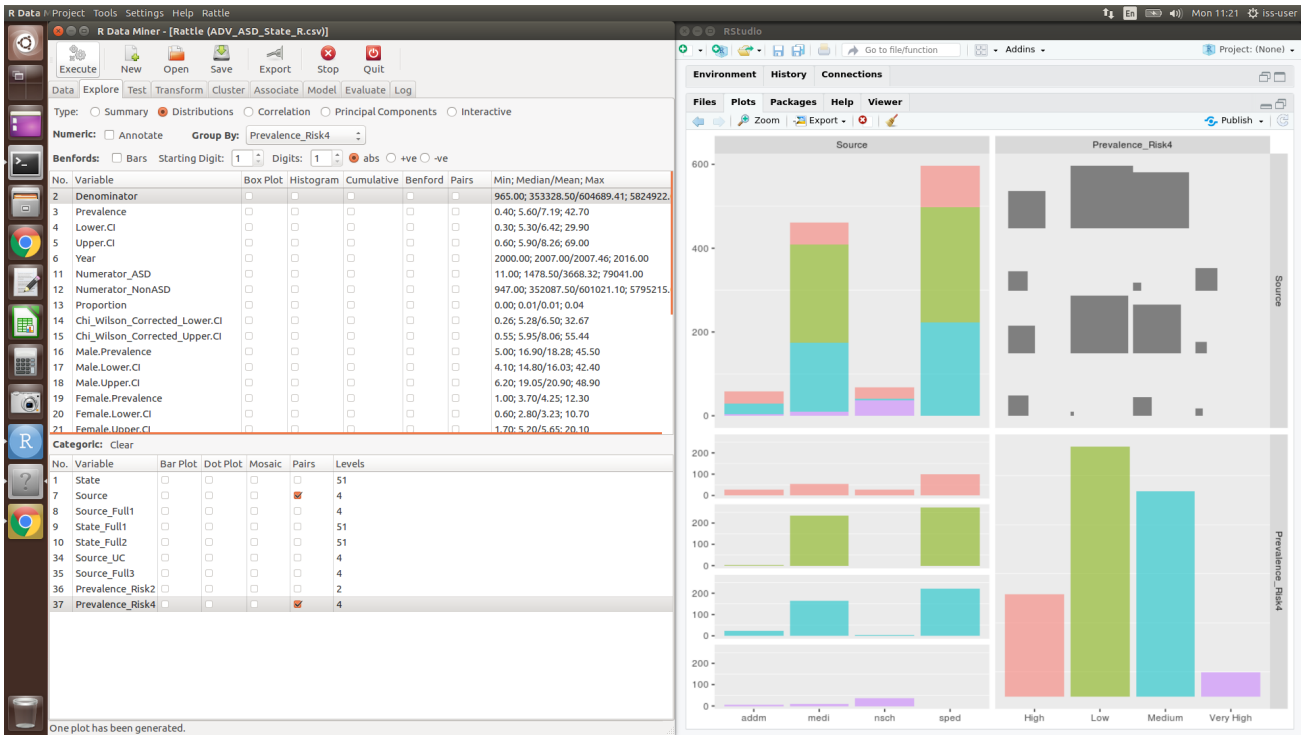
```
Loading required package: gplots

Attaching package: ‘gplots’

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

    lowess
```


R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate
Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922.
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215.
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 Source_UC 4
35 Source_Full3 4
36 Prevalence_Risk2 2
37 Prevalence_Risk4 4
One plot has been generated.
RStudio
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Distribution of Source (sample)
by Prevalence_Risk4
Frequency
0
100
200
300
400
500
600
700
596
99
274
223
0
461
53
234
165
9
68
27
0
5
36
59
29
1
25
4
All
High
Low
Medium
Very High
sped
medi
nsch
addm
Source
Rattle 2019-Dec-23 11:16:36 iss-user

In [25]:

```
#=====================================================================
# Rattle timestamp: 2019-12-23 11:16:35 x86_64-pc-linux-gnu

# The 'gplots' package provides the 'barplot2' function.

library(gplots, quietly=TRUE)

#=====================================================================
# Rattle timestamp: 2019-12-23 11:16:36 x86_64-pc-linux-gnu

# Bar Plot

# Generate the summary data for plotting.

ds <- rbind(summary(na.omit(crs$dataset[crs$train,]$Source)),
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence

# Sort the entries.

ord <- order(ds[1,], decreasing=TRUE)

# Plot the data.

bp <-  barplot2(ds[,ord], beside=TRUE, ylab="Frequency", xlab="Source", ylim=c

# Add the actual frequencies.

text(bp, ds[,ord]+24, ds[,ord])

# Add a legend to the plot.

legend("topright", bty="n", c("All","High","Low","Medium","Very High"),  fill=

# Add a title to the plot.

title(main="Distribution of Source (sample)\nby Prevalence_Risk4",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
```

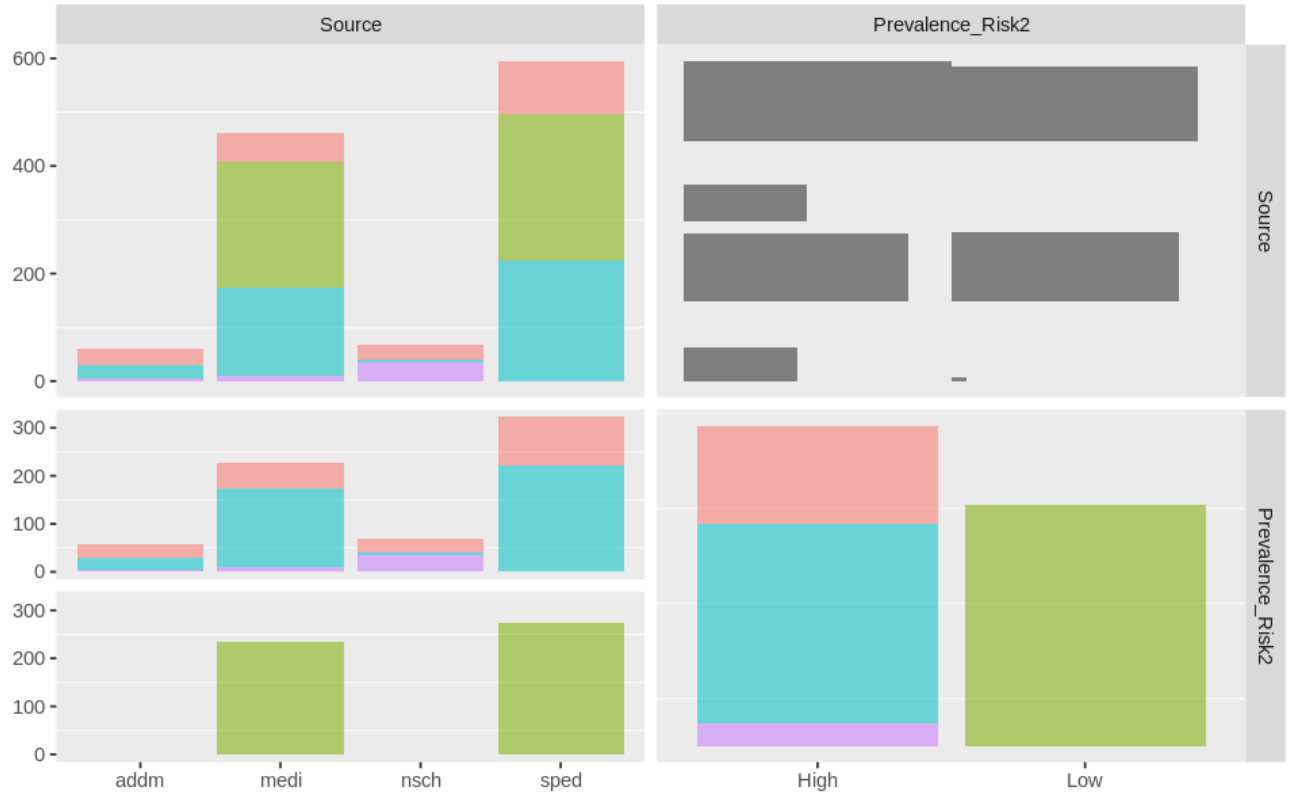

Distribution of Source (sample)
by Prevalence_Risk4
Frequency
0
100
200
300
400
500
600
700
596
99
274
223
0
461
53
234
165
9
68
27
0
5
36
59
29
1
25
4
All
High
Low
Medium
Very High
sped
medi
nsch
addm
Source
Rattle 2019-Dec-30 13:13:13 iss-user

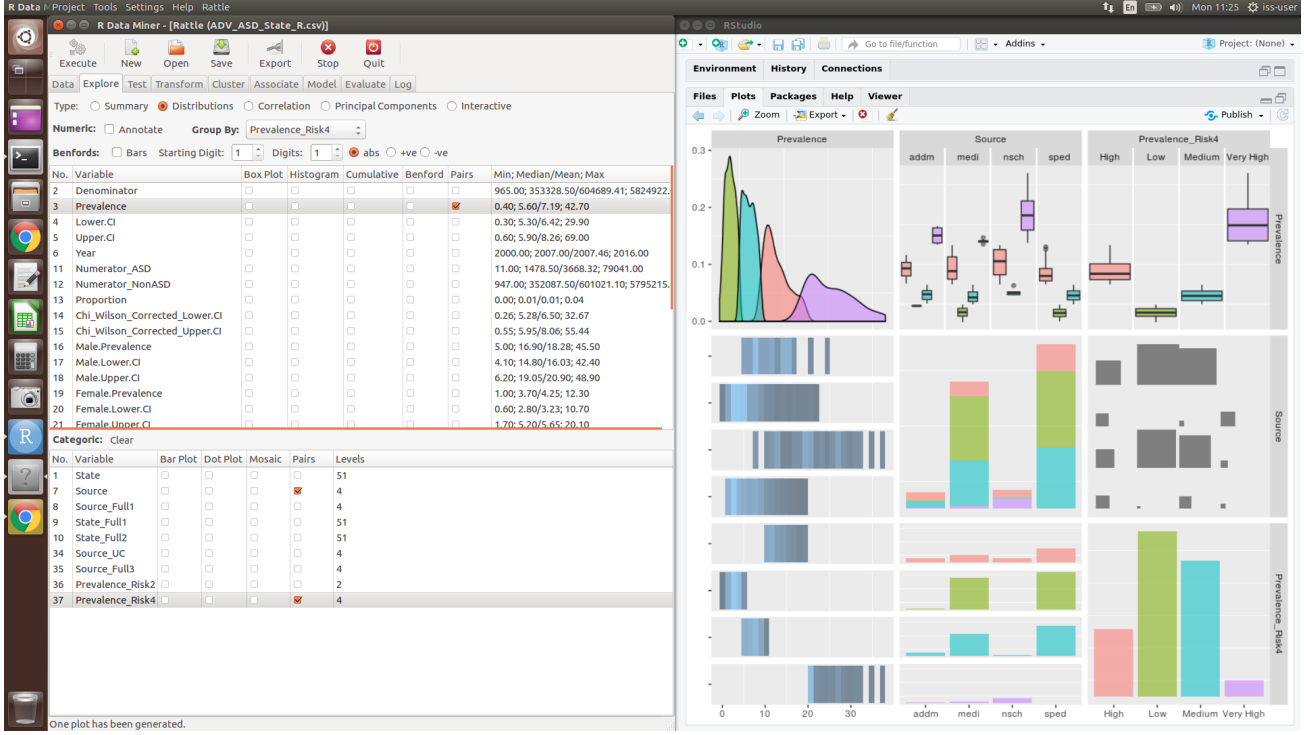

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate
Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922.
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215.
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 Source_UC 4
35 Source_Full3 4
36 Prevalence_Risk2 2
37 Prevalence_Risk4 4
One plot has been generated.
RStudio
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Source
Prevalence_Risk4
addm
medi
nsch
sped
High
Low
Medium
Very High

In [26]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:21:08 x86_64-pc-linux-gnu

# Display a pairs plot for the selected variables.

# Use GGally's ggpairs() to do the hard work.

crs$dataset[crs$train,] %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  GGally::ggpairs(columns=c(7,37),
        mapping=ggplot2::aes(colour=Prevalence_Risk4, alpha=0.5),
                diag=list(continuous="density",
                          discrete="bar"),
                upper=list(continuous="cor",
                           combo="box",
                           discrete="ratio"),
                lower=list(continuous="points",
                           combo="denstrip",
                           discrete="facetbar")) +
  ggplot2::theme(panel.grid.major=ggplot2::element_blank())
```

```
Warning message in check_and_set_ggpairs_defaults("diag", diag, continuous =
"densityDiag", :
“Changing diag$continuous from 'density' to 'densityDiag'”Warning message in
check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", :
“Changing diag$discrete from 'bar' to 'barDiag'”
```

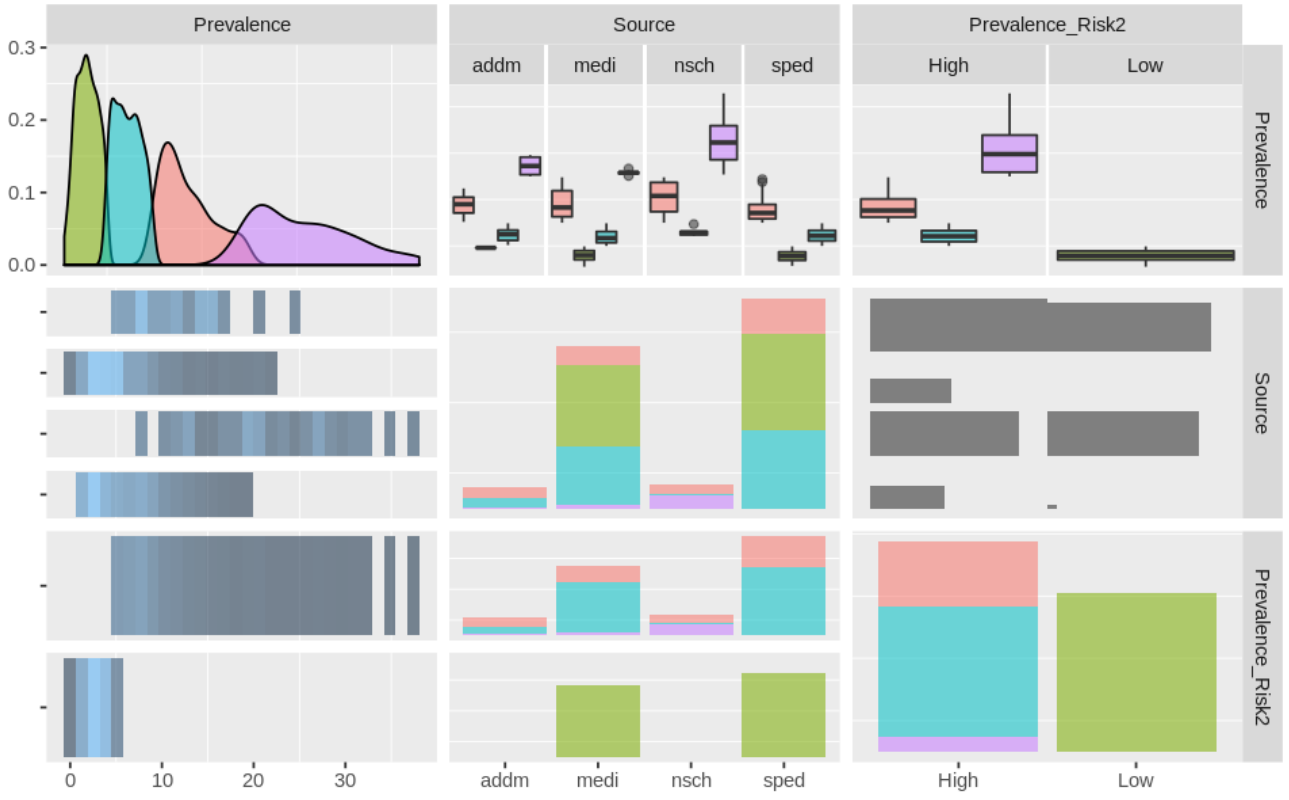



R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Summary Distributions Correlation Principal Components Interactive
Numeric: Annotate
Group By: Prevalence_Risk4
Benfords: Bars Starting Digit: 1 Digits: 1 abs +ve -ve
No. Variable Box Plot Histogram Cumulative Benford Pairs Min; Median/Mean; Max
2 Denominator 965.00; 353328.50/604689.41; 5824922.
3 Prevalence 0.40; 5.60/7.19; 42.70
4 Lower.CI 0.30; 5.30/6.42; 29.90
5 Upper.CI 0.60; 5.90/8.26; 69.00
6 Year 2000.00; 2007.00/2007.46; 2016.00
11 Numerator_ASD 11.00; 1478.50/3668.32; 79041.00
12 Numerator_NonASD 947.00; 352087.50/601021.10; 5795215.
13 Proportion 0.00; 0.01/0.01; 0.04
14 Chi_Wilson_Corrected_Lower.CI 0.26; 5.28/6.50; 32.67
15 Chi_Wilson_Corrected_Upper.CI 0.55; 5.95/8.06; 55.44
16 Male.Prevalence 5.00; 16.90/18.28; 45.50
17 Male.Lower.CI 4.10; 14.80/16.03; 42.40
18 Male.Upper.CI 6.20; 19.05/20.90; 48.90
19 Female.Prevalence 1.00; 3.70/4.25; 12.30
20 Female.Lower.CI 0.60; 2.80/3.23; 10.70
21 Female.Upper.CI 1.70; 5.20/5.65; 20.10
Categoric: Clear
No. Variable Bar Plot Dot Plot Mosaic Pairs Levels
1 State 51
7 Source 4
8 Source_Full1 4
9 State_Full1 51
10 State_Full2 51
34 Source_UC 4
35 Source_Full3 4
36 Prevalence_Risk2 2
37 Prevalence_Risk4 4
One plot has been generated.
RStudio
Addins
Project: (None)
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Prevalence
Source
Prevalence_Risk4
addm
medi
nsch
sped
High
Low
Medium
Very High
0.3
0.2
0.1
0.0
0
10
20
30

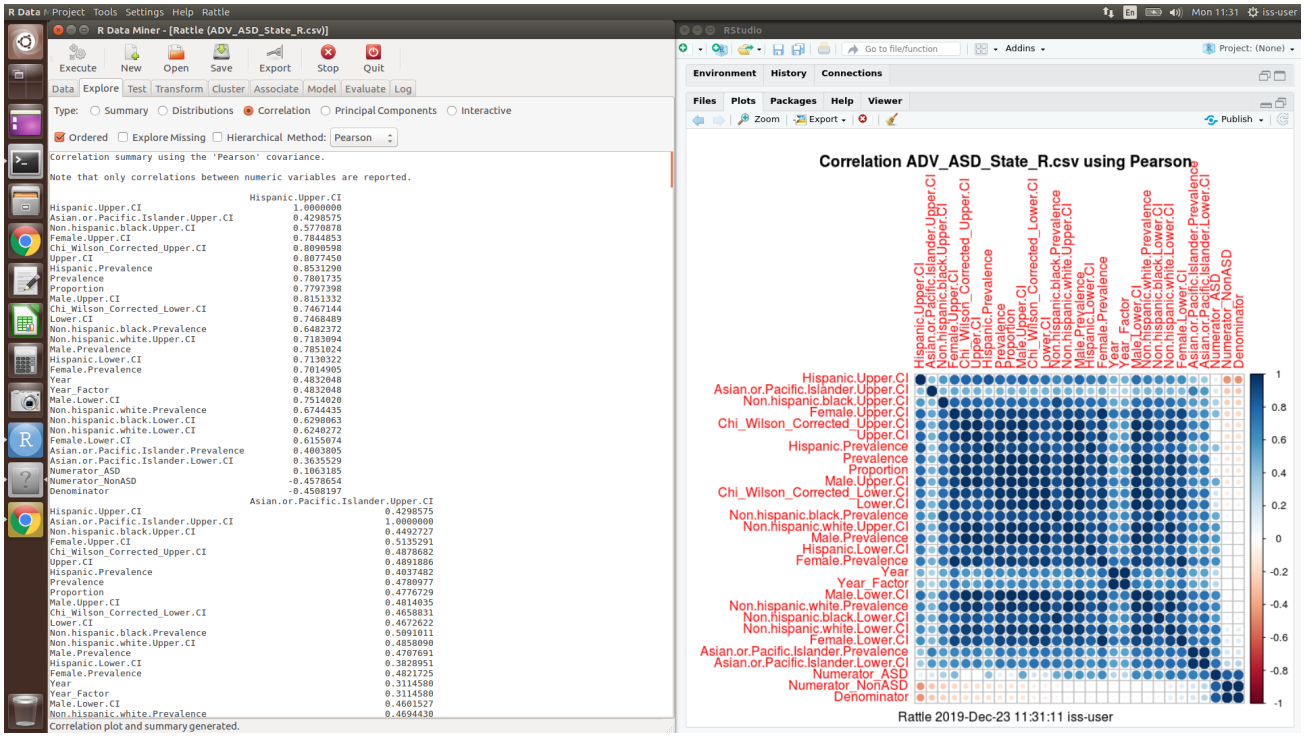
In [27]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:25:27 x86_64-pc-linux-gnu

# Display a pairs plot for the selected variables.

# Use GGally's ggpairs() to do the hard work.

crs$dataset[crs$train,] %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  GGally::ggpairs(columns=c(3,7,37),
        mapping=ggplot2::aes(colour=Prevalence_Risk4, alpha=0.5),
                diag=list(continuous="density",
                          discrete="bar"),
                upper=list(continuous="cor",
                           combo="box",
                           discrete="ratio"),
                lower=list(continuous="points",
                           combo="denstrip",
                           discrete="facetbar")) +
  ggplot2::theme(panel.grid.major=ggplot2::element_blank())
```

```
Warning message in check_and_set_ggpairs_defaults("diag", diag, continuous =
"densityDiag", :
“Changing diag$continuous from 'density' to 'densityDiag'”Warning message in
check_and_set_ggpairs_defaults("diag", diag, continuous = "densityDiag", :
“Changing diag$discrete from 'bar' to 'barDiag'”`stat_bin()` using `bins = 30
`. Pick better value with `binwidth`.
Warning message:
“Removed 54 rows containing missing values (geom_bar).”`stat_bin()` using `bi
ns = 30`. Pick better value with `binwidth`.
Warning message:
“Removed 31 rows containing missing values (geom_bar).”
```



Rattle: EDA Explore & Test: Correlation




```
                                  Hispanic.Upper.CI
Hispanic.Upper.CI                         1.0000000
Asian.or.Pacific.Islander.Upper.CI        0.4298575
Non.hispanic.black.Upper.CI               0.5770878
Female.Upper.CI                           0.7844853
Chi_Wilson_Corrected_Upper.CI             0.8090598
Upper.CI                                  0.8077450
Hispanic.Prevalence                       0.8531290
Prevalence                                0.7801735
Proportion                                0.7797398
Male.Upper.CI                             0.8151332
Chi_Wilson_Corrected_Lower.CI             0.7467144
Lower.CI                                  0.7468489
Non.hispanic.black.Prevalence             0.6482372
Non.hispanic.white.Upper.CI               0.7183094
Male.Prevalence                           0.7851024
Hispanic.Lower.CI                         0.7130322
Female.Prevalence                         0.7014905
Year                                      0.4832048
Year_Factor                               0.4832048
Male.Lower.CI                             0.7514020
Non.hispanic.white.Prevalence             0.6744435
Non.hispanic.black.Lower.CI               0.6298063
Non.hispanic.white.Lower.CI               0.6240272
Female.Lower.CI                           0.6155074
Asian.or.Pacific.Islander.Prevalence      0.4003805
Asian.or.Pacific.Islander.Lower.CI        0.3635529
Numerator_ASD                             0.1063185
Numerator_NonASD                         -0.4578654
Denominator                              -0.4508197
                                  Asian.or.Pacific.Islander.Upper.CI
Hispanic.Upper.CI                                          0.4298575
Asian.or.Pacific.Islander.Upper.CI                         1.0000000
Non.hispanic.black.Upper.CI                                0.4492727
Female.Upper.CI                                            0.5135291
Chi_Wilson_Corrected_Upper.CI                              0.4878682
Upper.CI                                                   0.4891886
Hispanic.Prevalence                                        0.4037482
Prevalence                                                 0.4780977
Proportion                                                 0.4776729
Male.Upper.CI                                              0.4814035
Chi_Wilson_Corrected_Lower.CI                              0.4658831
Lower.CI                                                   0.4672622
Non.hispanic.black.Prevalence                              0.5091011
Non.hispanic.white.Upper.CI                                0.4858090
Male.Prevalence                                            0.4707691
Hispanic.Lower.CI                                          0.3828951
Female.Prevalence                                          0.4821725
Year                                                       0.3114580
Year_Factor                                                0.3114580
Male.Lower.CI                                              0.4601527
Non.hispanic.white.Prevalence                              0.4694430
```

In [28]:

```
#=====================================================================
# Rattle timestamp: 2019-12-23 11:31:10 x86_64-pc-linux-gnu

# Generate a correlation plot for the variables.

# The 'corrplot' package provides the 'corrplot' function.

library(corrplot, quietly=TRUE)

# Correlations work for numeric variables only.

crs$cor <- cor(crs$dataset[crs$train, crs$numeric], use="pairwise", method="pe

# Order the correlations by their strength.

crs$ord <- order(crs$cor[1,])
crs$cor <- crs$cor[crs$ord, crs$ord]

# Display the actual correlations.

print(crs$cor)

# Graphically display the correlations.

corrplot(crs$cor, mar=c(0,0,1,0))
title(main="Correlation ADV_ASD_State_R.csv using Pearson",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
```

```
corrplot 0.84 loaded
```

R Data Project Tools Settings Help Rattle

Mon 11:32 iss-user

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]

Execute New Open Save Export Stop Quit

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: Summary Distributions Correlation Principal Components Interactive

Ordered Explore Missing Hierarchical Method: Pearson

```
Correlation summary using the 'Pearson' covariance.

Note that only correlations between numeric variables are reported.

                                  Hispanic.Upper.CI
Hispanic.Upper.CI                         1.0000000
Asian.or.Pacific.Islander.Upper.CI        0.4298575
Non.hispanic.black.Upper.CI               0.5770878
Female.Upper.CI                           0.7844853
Chi_Wilson_Corrected_Upper.CI             0.8090598
Upper.CI                                  0.8077450
Hispanic.Prevalence                       0.8531290
Prevalence                                0.7801735
Proportion                                0.7797398
Male.Upper.CI                             0.8151332
Chi_Wilson_Corrected_Lower.CI             0.7467144
Lower.CI                                  0.7468489
Non.hispanic.black.Prevalence             0.6482372
Non.hispanic.white.Upper.CI               0.7183094
Male.Prevalence                           0.7851024
Hispanic.Lower.CI                         0.7130322
Female.Prevalence                         0.7014905
Year                                      0.4832048
Year_Factor                               0.4832048
Male.Lower.CI                             0.7514020
Non.hispanic.white.Prevalence             0.6744435
Non.hispanic.black.Lower.CI               0.6298063
Non.hispanic.white.Lower.CI               0.6240272
Female.Lower.CI                           0.6155074
Asian.or.Pacific.Islander.Prevalence      0.4003805
Asian.or.Pacific.Islander.Lower.CI        0.3635529
Numerator_ASD                             0.1063185
Numerator_NonASD                         -0.4578654
Denominator                              -0.4508197
                                  Asian.or.Pacific.Islander.Upper.CI
Hispanic.Upper.CI                                          0.4298575
Asian.or.Pacific.Islander.Upper.CI                         1.0000000
Non.hispanic.black.Upper.CI                                0.4492727
Female.Upper.CI                                            0.5135291
Chi_Wilson_Corrected_Upper.CI                              0.4878682
Upper.CI                                                   0.4891886
Hispanic.Prevalence                                        0.4037482
Prevalence                                                 0.4780977
Proportion                                                 0.4776729
Male.Upper.CI                                              0.4814035
Chi_Wilson_Corrected_Lower.CI                              0.4658831
Lower.CI                                                   0.4672622
Non.hispanic.black.Prevalence                              0.5091011
Non.hispanic.white.Upper.CI                                0.4858090
Male.Prevalence                                            0.4707691
Hispanic.Lower.CI                                          0.3828951
Female.Prevalence                                          0.4821725
Year                                                       0.3114580
Year_Factor                                                0.3114580
Male.Lower.CI                                              0.4601527
Non.hispanic.white.Prevalence                              0.4694430
```

Hierarchical cluster of correlations plotted.

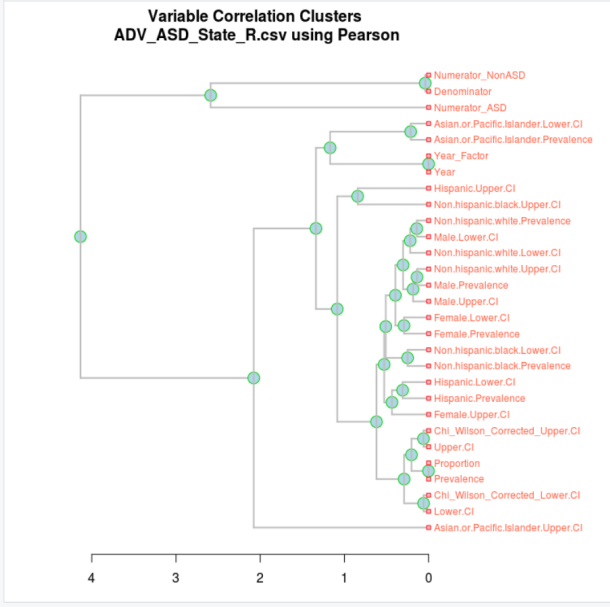
RStudio

Go to file/function Addins Project: (None)

Environment History Connections

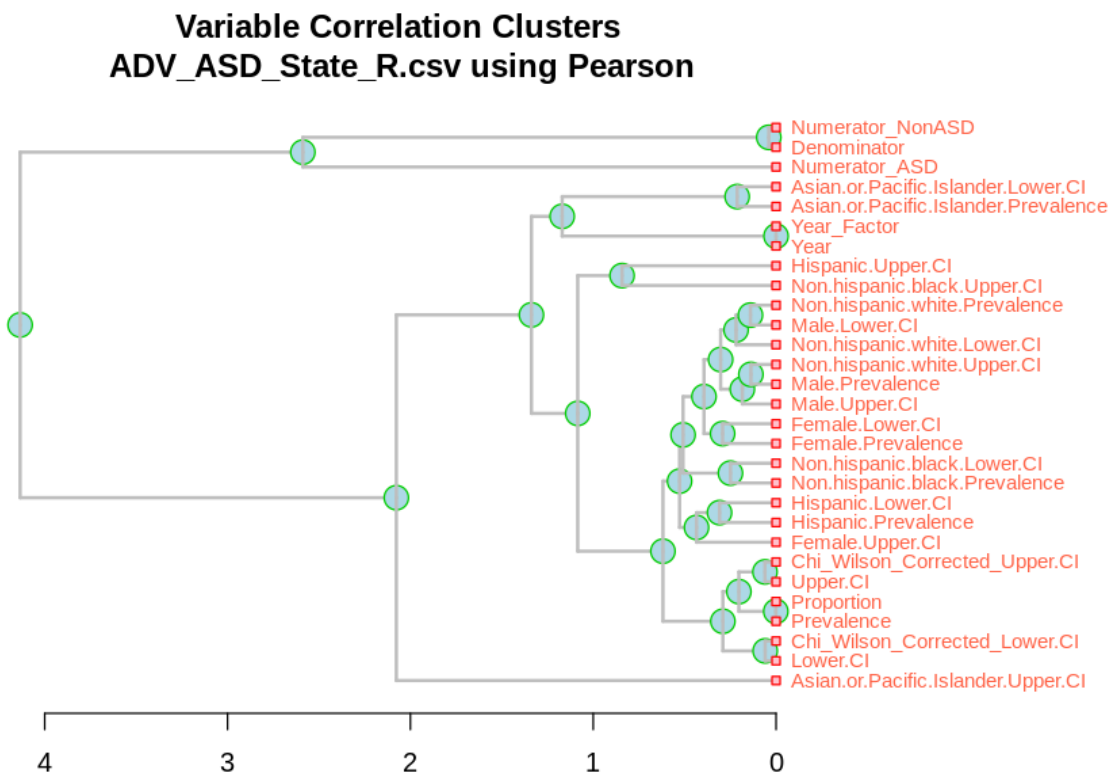
Files Plots Packages Help Viewer

Zoom Export Publish



In [29]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:32:46 x86_64-pc-linux-gnu

# Hierarchical Variable Correlation

# Generate the correlations (numerics only).

cc <- cor(crs$dataset[crs$train, crs$numeric], use="pairwise", method="pearson

# Generate hierarchical cluster of variables.

hc <- hclust(dist(cc), method="average")

# Generate the dendrogram.

dn <- as.dendrogram(hc)

# Now draw the dendrogram.

op <- par(mar = c(3, 4, 3, 10.86))
plot(dn, horiz = TRUE, nodePar = list(col = 3:2, cex = c(2.0, 0.75), pch = 21:
title(main="Variable Correlation Clusters
 ADV_ASD_State_R.csv using Pearson",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
par(op)
```



R Data Project Tools Settings Help Rattle

Mon 11:34 iss-user

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]

Execute New Open Save Export Stop Quit

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: Summary Distributions Correlation Principal Components Interactive

Ordered Explore Missing Hierarchical Method: Spearman

```
Correlation summary using the 'Pearson' covariance.

Note that only correlations between numeric variables are reported.

                                    Hispanic.Upper.CI
Hispanic.Upper.CI                           1.0000000
Asian.or.Pacific.Islander.Upper.CI          0.4298575
Non.hispanic.black.Upper.CI                 0.5770878
Female.Upper.CI                             0.7844853
Chi_Wilson_Corrected_Upper.CI               0.8090598
Upper.CI                                    0.8077450
Hispanic.Prevalence                         0.8531290
Prevalence                                  0.7801735
Proportion                                  0.7797398
Male.Upper.CI                               0.8151332
Chi_Wilson_Corrected_Lower.CI               0.7467144
Lower.CI                                    0.7468489
Non.hispanic.black.Prevalence               0.6482372
Non.hispanic.white.Upper.CI                 0.7183094
Male.Prevalence                             0.7851024
Hispanic.Lower.CI                           0.7130322
Female.Prevalence                           0.7014905
Year                                        0.4832048
Year_Factor                                 0.4832048
Male.Lower.CI                               0.7514020
Non.hispanic.white.Prevalence               0.6744435
Non.hispanic.black.Lower.CI                 0.6298063
Non.hispanic.white.Lower.CI                 0.6240272
Female.Lower.CI                             0.6155074
Asian.or.Pacific.Islander.Prevalence        0.4003805
Asian.or.Pacific.Islander.Lower.CI          0.3635529
Numerator_ASD                               0.1063185
Numerator_NonASD                           -0.4578654
Denominator                                -0.4508197
                                    Asian.or.Pacific.Islander.Upper.CI
Hispanic.Upper.CI                                           0.4298575
Asian.or.Pacific.Islander.Upper.CI                          1.0000000
Non.hispanic.black.Upper.CI                                 0.4492727
Female.Upper.CI                                             0.5135291
Chi_Wilson_Corrected_Upper.CI                               0.4878682
Upper.CI                                                    0.4891886
Hispanic.Prevalence                                         0.4037482
Prevalence                                                  0.4780977
Proportion                                                  0.4776729
Male.Upper.CI                                               0.4814035
Chi_Wilson_Corrected_Lower.CI                               0.4658831
Lower.CI                                                    0.4672622
Non.hispanic.black.Prevalence                               0.5091011
Non.hispanic.white.Upper.CI                                 0.4858090
Male.Prevalence                                             0.4707691
Hispanic.Lower.CI                                           0.3828951
Female.Prevalence                                           0.4821725
Year                                                        0.3114580
Year_Factor                                                 0.3114580
Male.Lower.CI                                               0.4601527
Non.hispanic.white.Prevalence                               0.4694430
```

Hierarchical cluster of correlations plotted.

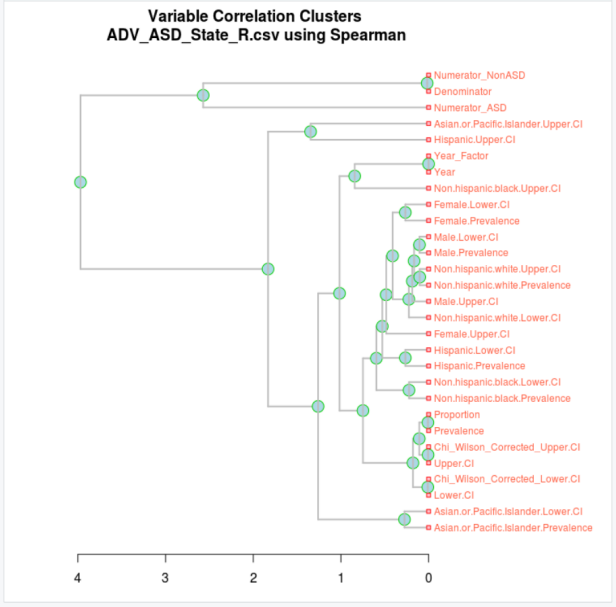
RStudio

Go to file/function Addins Project: (None)

Environment History Connections

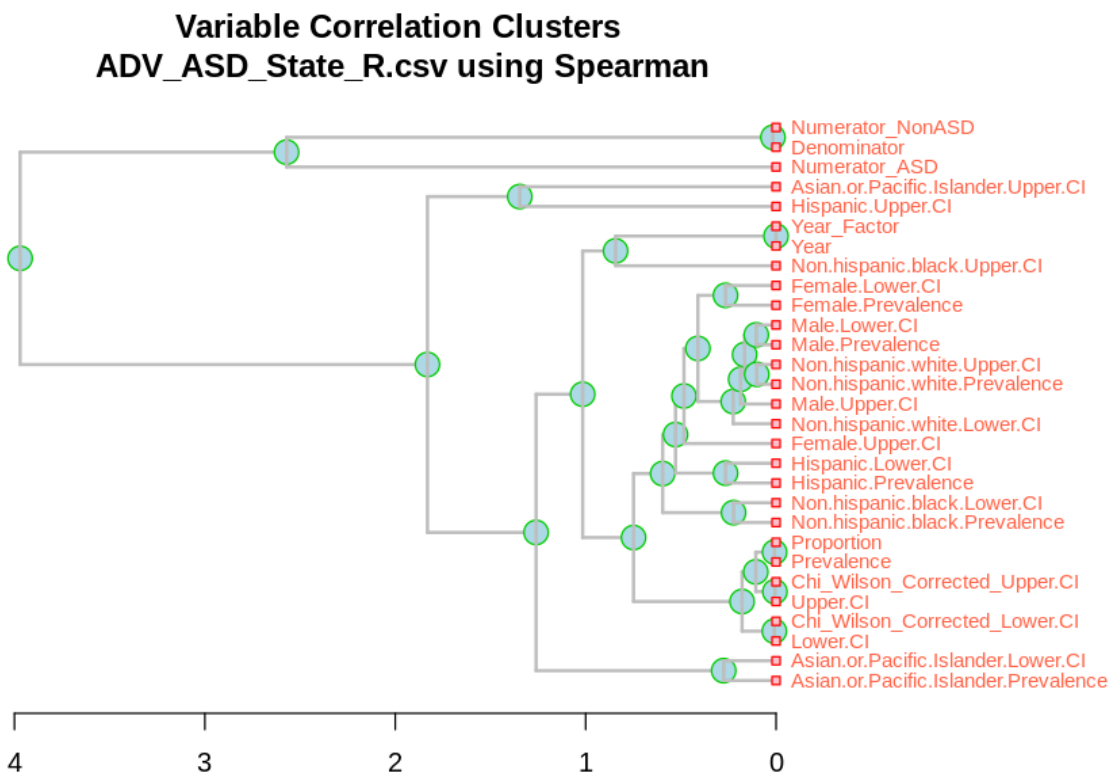
Files Plots Packages Help Viewer

Zoom Export Publish



In [30]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:33:55 x86_64-pc-linux-gnu

# Hierarchical Variable Correlation

# Generate the correlations (numerics only).

cc <- cor(crs$dataset[crs$train, crs$numeric], use="pairwise", method="spearma

# Generate hierarchical cluster of variables.

hc <- hclust(dist(cc), method="average")

# Generate the dendrogram.

dn <- as.dendrogram(hc)

# Now draw the dendrogram.

op <- par(mar = c(3, 4, 3, 10.86))
plot(dn, horiz = TRUE, nodePar = list(col = 3:2, cex = c(2.0, 0.75), pch = 21:
title(main="Variable Correlation Clusters
 ADV_ASD_State_R.csv using Spearman",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
par(op)
```



R Data Project Tools Settings Help Rattle

Mon 11:36 iss-user

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]

Execute New Open Save Export Stop Quit

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: Summary Distributions Correlation Principal Components Interactive

Method: SVD Eigen

```
Note that principal components on only the numeric
 variables is calculated, and so we can not use this
 approach to remove categoric variables from  consideration.

 Any numeric variables with relatively large rotation
 values (negative or positive) in any of the first few
 components are generally variables that you may wish
 to include in the modelling.

Rattle timestamp: 2019-12-23 11:36:12 iss-user

============================================================================
Standard deviations (1, .., p=29):
 [1] 4.656380e+00 1.820987e+00
 [3] 1.112265e+00 9.545395e-01
 [5] 7.437407e-01 6.229362e-01
 [7] 5.188450e-01 4.828164e-01
 [9] 4.237121e-01 3.097548e-01
[11] 2.296600e-01 2.174858e-01
[13] 1.414526e-01 8.282109e-02
[15] 6.362457e-02 4.164272e-02
[17] 2.967576e-02 2.474057e-02
[19] 1.782267e-02 1.467028e-02
[21] 9.427596e-03 4.767943e-03
[23] 3.824069e-03 2.852320e-03
[25] 1.584068e-03 1.408651e-03
[27] 5.042372e-04 5.196008e-16
[29] 4.246809e-16

Rotation (n x k) = (29 x 29):
                                                PC1
Denominator                            0.01883922
Prevalence                            -0.21289953
Lower.CI                              -0.21278276
Upper.CI                              -0.21143642
Year                                  -0.16313959
Numerator_ASD                         -0.12486026
Numerator_NonASD                       0.02110653
Proportion                            -0.21289857
Chi_Wilson_Corrected_Lower.CI         -0.21274240
Chi_Wilson_Corrected_Upper.CI         -0.21131533
Male.Prevalence                       -0.21126548
Male.Lower.CI                         -0.21147726
Male.Upper.CI                         -0.20911867
Female.Prevalence                     -0.20491117
Female.Lower.CI                       -0.19893453
Female.Upper.CI                       -0.20270439
Non.hispanic.white.Prevalence         -0.20854472
Non.hispanic.white.Lower.CI           -0.20668534
Non.hispanic.white.Upper.CI           -0.20772011
Non.hispanic.black.Prevalence         -0.19617202
Non.hispanic.black.Lower.CI           -0.19374133
Non.hispanic.black.Upper.CI           -0.17164747
Hispanic.Prevalence                   -0.19507691
Hispanic.Lower.CI                     -0.19350606
Hispanic.Upper.CI                     -0.16406062
Asian.or.Pacific.Islander.Prevalence  -0.16511852
```

A principal components analysis has been completed.

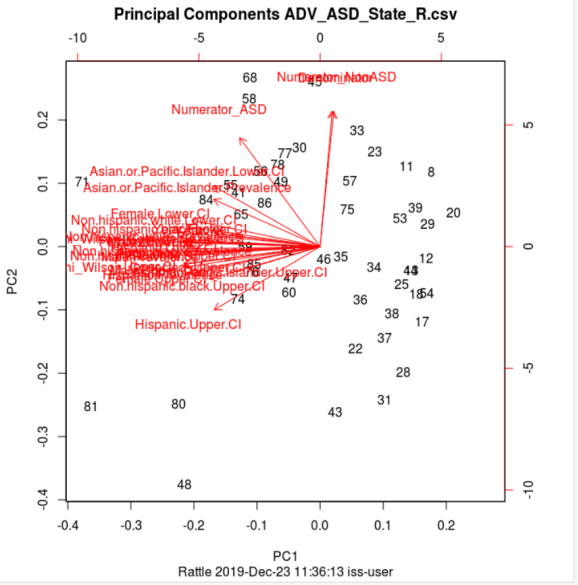
RStudio

Go to file/function Addins Project: (None)

Environment History Connections

Files Plots Packages Help Viewer

Zoom Export Publish

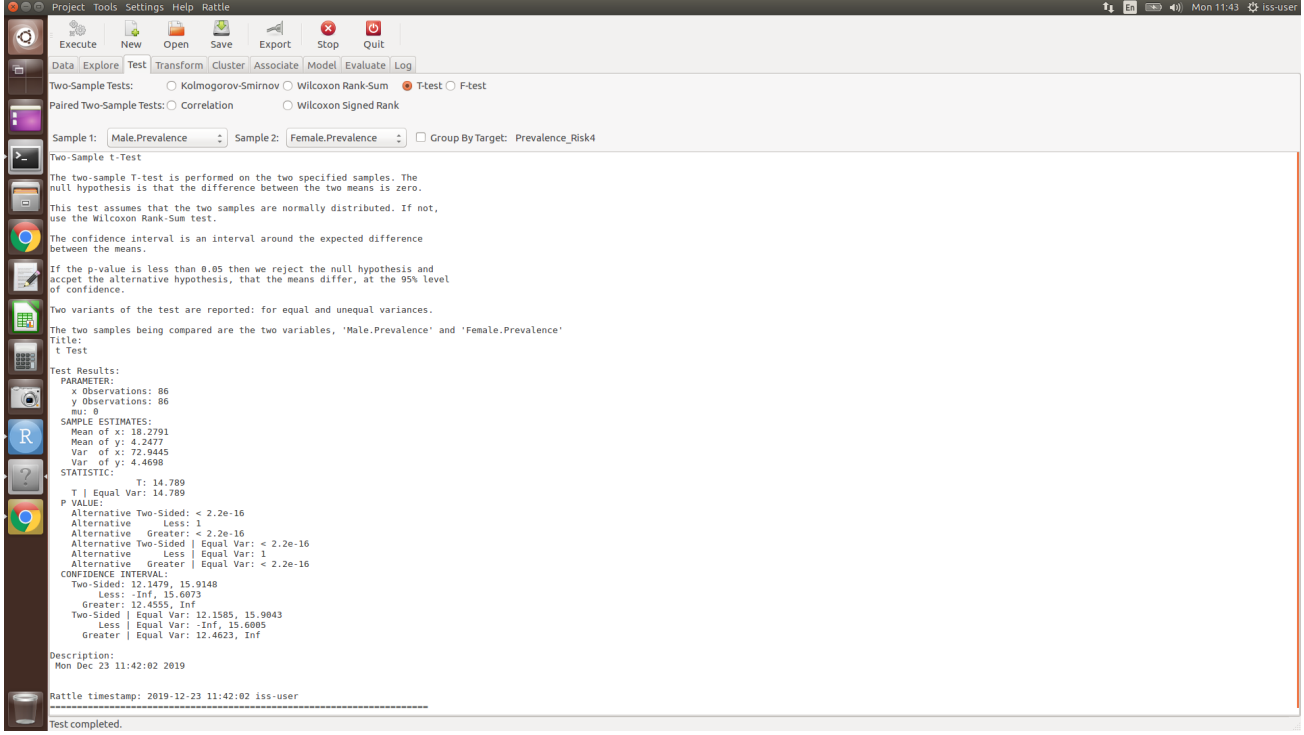


In [31]:

```
#=======================================================================
# Rattle timestamp: 2019-12-23 11:36:12 x86_64-pc-linux-gnu

# Principal Components Analysis (on numerics only).

pc <- prcomp(na.omit(crs$dataset[crs$train, crs$numeric]), scale=TRUE, center=

# Show the output of the analysis.

pc

# Summarise the importance of the components found.

summary(pc)

# Display a plot showing the relative importance of the components.

plot(pc, main="")
title(main="Principal Components Importance ADV_ASD_State_R.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
axis(1, at=seq(0.7, ncol(pc$rotation)*1.2, 1.2), labels=colnames(pc$rotation),

# Display a plot showing the two most principal components.

biplot(pc, main="")
title(main="Principal Components ADV_ASD_State_R.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
```

```
Standard deviations (1, .., p=29):
 [1] 4.656380e+00 1.820987e+00 1.112265e+00 9.545395e-01 7.437407e-01
 [6] 6.229362e-01 5.188450e-01 4.828164e-01 4.237121e-01 3.097548e-01
[11] 2.296600e-01 2.174858e-01 1.414526e-01 8.282109e-02 6.362457e-02
[16] 4.164272e-02 2.967576e-02 2.474057e-02 1.782267e-02 1.467028e-02
[21] 9.427596e-03 4.767943e-03 3.824069e-03 2.852320e-03 1.504068e-03
[26] 1.408651e-03 5.042372e-04 4.246809e-16 1.799791e-16

Rotation (n x k) = (29 x 29):
                                          PC1          PC2          PC3
Denominator                        0.01883922  0.534519219 -0.085620632
Prevalence                        -0.21289953 -0.023107040 -0.044421446
Lower.CI                          -0.21278276  0.019630807 -0.046621564
Upper.CI                          -0.21143642 -0.067635880 -0.037749486
Year                              -0.16313959  0.052170196 -0.295735159
Numerator_ASD                     -0.12486026  0.429738087 -0.033616427
Numerator_NonASD                   0.02110653  0.533643140 -0.086058432
Proportion                        -0.21289857 -0.022459211 -0.044693299
Chi_Wilson_Corrected_Lower.CI     -0.21274240  0.020491372 -0.047530630
```

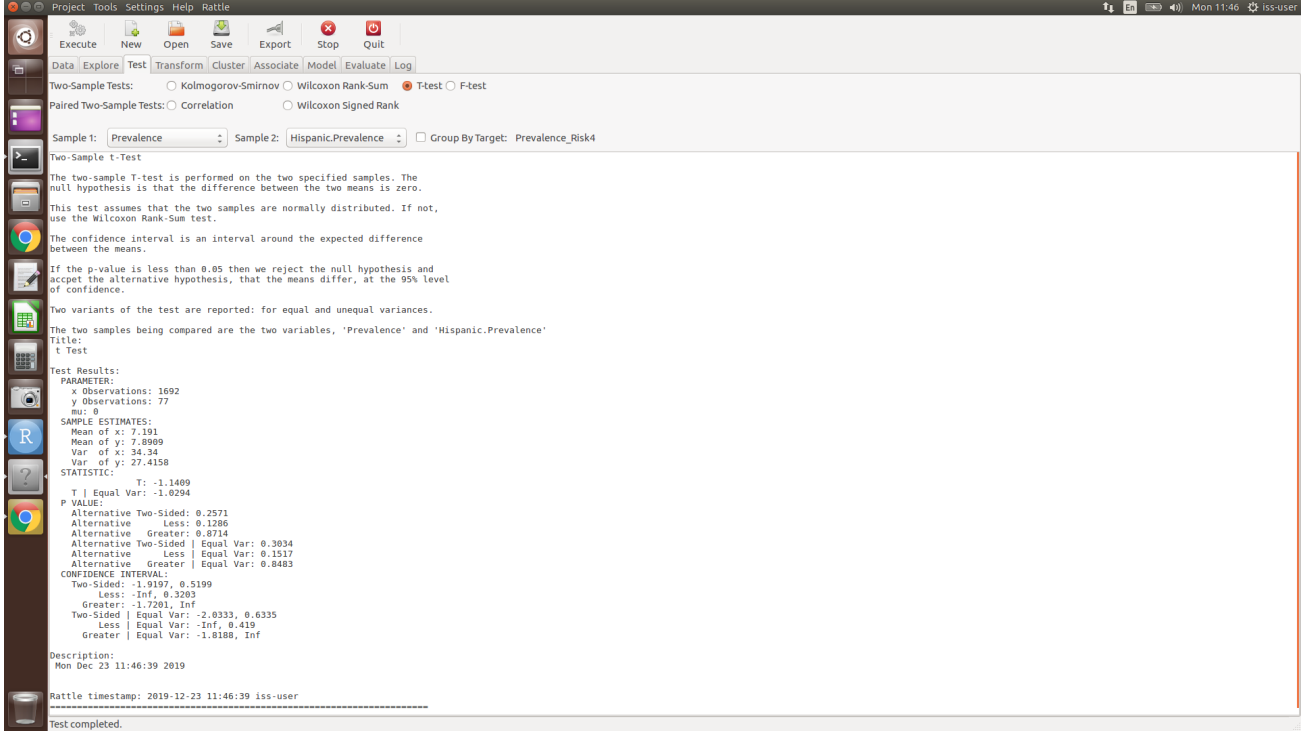
Rattle: EDA Explore & Test: Test


Project Tools Settings Help Rattle
Mon 11:43 iss-user
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Two-Sample Tests: Kolmogorov-Smirnov Wilcoxon Rank-Sum T-test F-test
Paired Two-Sample Tests: Correlation Wilcoxon Signed Rank
Sample 1: Male.Prevalence Sample 2: Female.Prevalence Group By Target: Prevalence_Risk4
Two-Sample t-Test
The two-sample T-test is performed on the two specified samples. The
null hypothesis is that the difference between the two means is zero.
This test assumes that the two samples are normally distributed. If not,
use the Wilcoxon Rank-Sum test.
The confidence interval is an interval around the expected difference
between the means.
If the p-value is less than 0.05 then we reject the null hypothesis and
accpet the alternative hypothesis, that the means differ, at the 95% level
of confidence.
Two variants of the test are reported: for equal and unequal variances.
The two samples being compared are the two variables, 'Male.Prevalence' and 'Female.Prevalence'
Title:
t Test
Test Results:
PARAMETER:
x Observations: 86
y Observations: 86
mu: 0
SAMPLE ESTIMATES:
Mean of x: 18.2791
Mean of y: 4.2477
Var of x: 72.9445
Var of y: 4.4698
STATISTIC:
T: 14.789
T | Equal Var: 14.789
P VALUE:
Alternative Two-Sided: < 2.2e-16
Alternative Less: 1
Alternative Greater: < 2.2e-16
Alternative Two-Sided | Equal Var: < 2.2e-16
Alternative Less | Equal Var: 1
Alternative Greater | Equal Var: < 2.2e-16
CONFIDENCE INTERVAL:
Two-Sided: 12.1479, 15.9148
Less: -Inf, 15.6073
Greater: 12.4555, Inf
Two-Sided | Equal Var: 12.1585, 15.9043
Less | Equal Var: -Inf, 15.6005
Greater | Equal Var: 12.4623, Inf
Description:
Mon Dec 23 11:42:02 2019
Rattle timestamp: 2019-12-23 11:42:02 iss-user
Test completed.

In [32]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:42:02 x86_64-pc-linux-gnu

# Perform Test

# Use the fBasics package for statistical tests.

library(fBasics, quietly=TRUE)

# Perform the test.

locationTest(na.omit(crs$dataset[, "Male.Prevalence"]), na.omit(crs$dataset[,
```

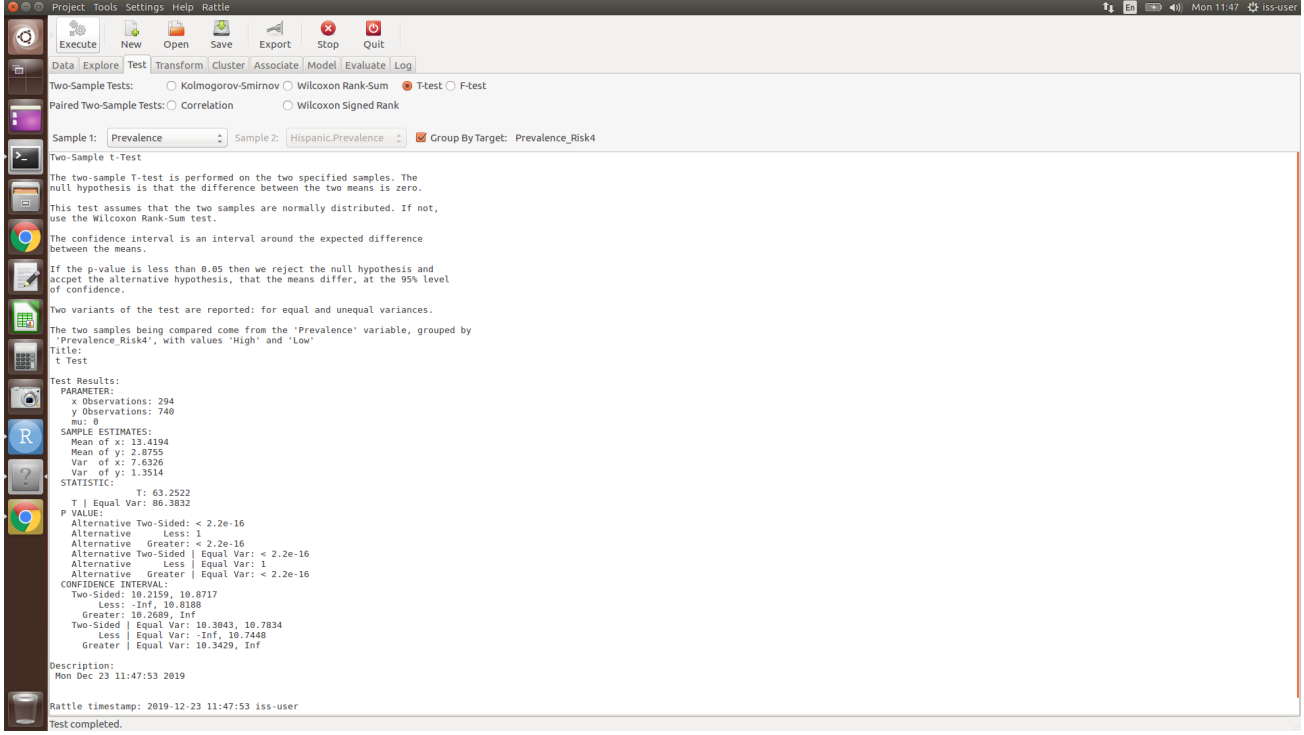
```
Title:
 t Test

Test Results:
  PARAMETER:
    x Observations: 86
    y Observations: 86
    mu: 0
  SAMPLE ESTIMATES:
    Mean of x: 18.2791
    Mean of y: 4.2477
    Var  of x: 72.9445
    Var  of y: 4.4698
  STATISTIC:
                 T: 14.789
    T | Equal Var: 14.789
  P VALUE:
    Alternative Two-Sided: < 2.2e-16
    Alternative      Less: 1
    Alternative   Greater: < 2.2e-16
    Alternative Two-Sided | Equal Var: < 2.2e-16
    Alternative      Less | Equal Var: 1
    Alternative   Greater | Equal Var: < 2.2e-16
  CONFIDENCE INTERVAL:
    Two-Sided: 12.1479, 15.9148
         Less: -Inf, 15.6073
      Greater: 12.4555, Inf
    Two-Sided | Equal Var: 12.1585, 15.9043
         Less | Equal Var: -Inf, 15.6005
      Greater | Equal Var: 12.4623, Inf

Description:
 Mon Dec 30 13:13:21 2019
```


Project Tools Settings Help Rattle
Mon 11:46 iss-user
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Two-Sample Tests: Kolmogorov-Smirnov Wilcoxon Rank-Sum T-test F-test
Paired Two-Sample Tests: Correlation Wilcoxon Signed Rank
Sample 1: Prevalence Sample 2: Hispanic.Prevalence Group By Target: Prevalence_Risk4
Two-Sample t-Test
The two-sample T-test is performed on the two specified samples. The
null hypothesis is that the difference between the two means is zero.
This test assumes that the two samples are normally distributed. If not,
use the Wilcoxon Rank-Sum test.
The confidence interval is an interval around the expected difference
between the means.
If the p-value is less than 0.05 then we reject the null hypothesis and
accpet the alternative hypothesis, that the means differ, at the 95% level
of confidence.
Two variants of the test are reported: for equal and unequal variances.
The two samples being compared are the two variables, 'Prevalence' and 'Hispanic.Prevalence'
Title:
t Test
Test Results:
PARAMETER:
x Observations: 1692
y Observations: 77
mu: 0
SAMPLE ESTIMATES:
Mean of x: 7.191
Mean of y: 7.8909
Var of x: 34.34
Var of y: 27.4158
STATISTIC:
T: -1.1409
T | Equal Var: -1.0294
P VALUE:
Alternative Two-Sided: 0.2571
Alternative Less: 0.1286
Alternative Greater: 0.8714
Alternative Two-Sided | Equal Var: 0.3034
Alternative Less | Equal Var: 0.1517
Alternative Greater | Equal Var: 0.8483
CONFIDENCE INTERVAL:
Two-Sided: -1.9197, 0.5199
Less: -Inf, 0.3203
Greater: -1.7201, Inf
Two-Sided | Equal Var: -2.0333, 0.6335
Less | Equal Var: -Inf, 0.419
Greater | Equal Var: -1.8188, Inf
Description:
Mon Dec 23 11:46:39 2019
Rattle timestamp: 2019-12-23 11:46:39 iss-user
Test completed.

In [33]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:46:39 x86_64-pc-linux-gnu

# Perform Test

# Use the fBasics package for statistical tests.

library(fBasics, quietly=TRUE)

# Perform the test.

locationTest(na.omit(crs$dataset[, "Prevalence"]), na.omit(crs$dataset[, "Hisp
```

```
Title:
 t Test

Test Results:
  PARAMETER:
    x Observations: 1692
    y Observations: 77
    mu: 0
  SAMPLE ESTIMATES:
    Mean of x: 7.191
    Mean of y: 7.8909
    Var  of x: 34.34
    Var  of y: 27.4158
  STATISTIC:
                  T: -1.1409
    T | Equal Var: -1.0294
  P VALUE:
    Alternative Two-Sided: 0.2571
    Alternative      Less: 0.1286
    Alternative   Greater: 0.8714
    Alternative Two-Sided | Equal Var: 0.3034
    Alternative      Less | Equal Var: 0.1517
    Alternative   Greater | Equal Var: 0.8483
  CONFIDENCE INTERVAL:
    Two-Sided: -1.9197, 0.5199
         Less: -Inf, 0.3203
      Greater: -1.7201, Inf
    Two-Sided | Equal Var: -2.0333, 0.6335
         Less | Equal Var: -Inf, 0.419
      Greater | Equal Var: -1.8188, Inf

Description:
 Mon Dec 30 13:13:21 2019
```


Project Tools Settings Help Rattle
Mon 11:47 iss-user
Execute
New
Open
Save
Export
Stop
Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Two-Sample Tests: Kolmogorov-Smirnov Wilcoxon Rank-Sum T-test F-test
Paired Two-Sample Tests: Correlation Wilcoxon Signed Rank
Sample 1: Prevalence
Sample 2: Hispanic.Prevalence
Group By Target: Prevalence_Risk4
Two-Sample t-Test
The two-sample T-test is performed on the two specified samples. The
null hypothesis is that the difference between the two means is zero.
This test assumes that the two samples are normally distributed. If not,
use the Wilcoxon Rank-Sum test.
The confidence interval is an interval around the expected difference
between the means.
If the p-value is less than 0.05 then we reject the null hypothesis and
accpet the alternative hypothesis, that the means differ, at the 95% level
of confidence.
Two variants of the test are reported: for equal and unequal variances.
The two samples being compared come from the 'Prevalence' variable, grouped by
'Prevalence_Risk4', with values 'High' and 'Low'
Title:
t Test
Test Results:
PARAMETER:
x Observations: 294
y Observations: 740
mu: 0
SAMPLE ESTIMATES:
Mean of x: 13.4194
Mean of y: 2.8755
Var of x: 7.6326
Var of y: 1.3514
STATISTIC:
T: 63.2522
T | Equal Var: 86.3832
P VALUE:
Alternative Two-Sided: < 2.2e-16
Alternative Less: 1
Alternative Greater: < 2.2e-16
Alternative Two-Sided | Equal Var: < 2.2e-16
Alternative Less | Equal Var: 1
Alternative Greater | Equal Var: < 2.2e-16
CONFIDENCE INTERVAL:
Two-Sided: 10.2159, 10.8717
Less: -Inf, 10.8188
Greater: 10.2689, Inf
Two-Sided | Equal Var: 10.3043, 10.7834
Less | Equal Var: -Inf, 10.7448
Greater | Equal Var: 10.3429, Inf
Description:
Mon Dec 23 11:47:53 2019
Rattle timestamp: 2019-12-23 11:47:53 iss-user
Test completed.

In [34]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 11:47:53 x86_64-pc-linux-gnu

# Perform Test

# Use the fBasics package for statistical tests.

library(fBasics, quietly=TRUE)

# Perform the test.

locationTest(na.omit(crs$dataset[crs$dataset[["Prevalence_Risk4"]] == "High",
```

```
Title:
 t Test

Test Results:
  PARAMETER:
    x Observations: 294
    y Observations: 740
    mu: 0
  SAMPLE ESTIMATES:
    Mean of x: 13.4194
    Mean of y: 2.8755
    Var  of x: 7.6326
    Var  of y: 1.3514
  STATISTIC:
                 T: 63.2522
    T | Equal Var: 86.3832
  P VALUE:
    Alternative Two-Sided: < 2.2e-16
    Alternative      Less: 1
    Alternative   Greater: < 2.2e-16
    Alternative Two-Sided | Equal Var: < 2.2e-16
    Alternative      Less | Equal Var: 1
    Alternative   Greater | Equal Var: < 2.2e-16
  CONFIDENCE INTERVAL:
    Two-Sided: 10.2159, 10.8717
         Less: -Inf, 10.8188
      Greater: 10.2689, Inf
    Two-Sided | Equal Var: 10.3043, 10.7834
         Less | Equal Var: -Inf, 10.7448
      Greater | Equal Var: 10.3429, Inf

Description:
 Mon Dec 30 13:13:21 2019
```

()

Rattle: Process & Transform Data

Rattle: Process & Transform Data: Select Variables

In [35]:
```
names(crs$dataset)
```

'State' 'Denominator' 'Prevalence' 'Lower.CI' 'Upper.CI' 'Year' 'Source' 'Source_Full1' 'State_Full1' 'State_Full2' 'Numerator_ASD' 'Numerator_NonASD' 'Proportion' 'Chi_Wilson_Corrected_Lower.CI' 'Chi_Wilson_Corrected_Upper.CI' 'Male.Prevalence' 'Male.Lower.CI' 'Male.Upper.CI' 'Female.Prevalence' 'Female.Lower.CI' 'Female.Upper.CI' 'Non.hispanic.white.Prevalence' 'Non.hispanic.white.Lower.CI' 'Non.hispanic.white.Upper.CI' 'Non.hispanic.black.Prevalence' 'Non.hispanic.black.Lower.CI' 'Non.hispanic.black.Upper.CI' 'Hispanic.Prevalence' 'Hispanic.Lower.CI' 'Hispanic.Upper.CI' 'Asian.or.Pacific.Islander.Prevalence' 'Asian.or.Pacific.Islander.Lower.CI' 'Asian.or.Pacific.Islander.Upper.CI' 'State_Region' 'Source_UC' 'Source_Full3' 'Prevalence_Risk2' 'Prevalence_Risk4' 'Year_Factor'

Select variables to build classification model

Target:

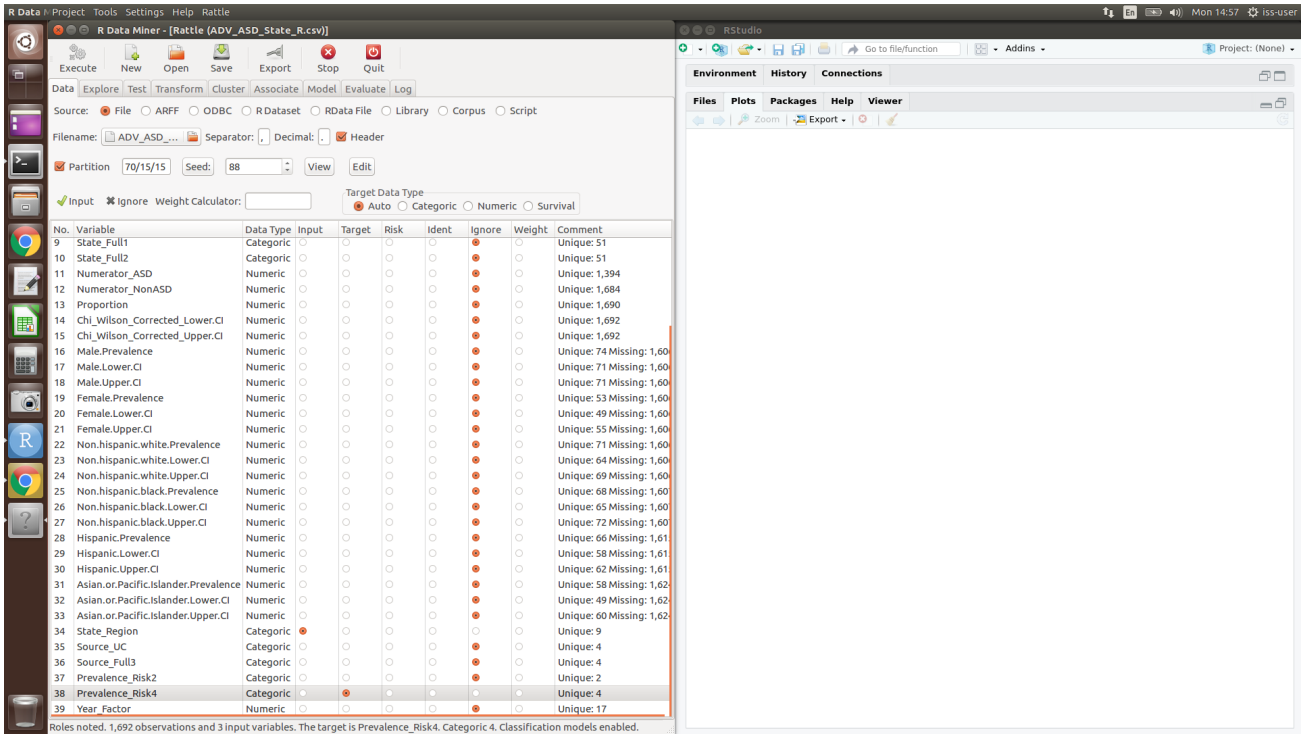
- Prevalence_Risk4

Inputs:

- Denominator
- Year
- Source
- State_Region

Note: It's not recommended ot use **State**, which has more than 32 levels.

https://stats.stackexchange.com/questions/49243/rs-randomforest-can-not-handle-more-than-32-levels-what-is-workaround (https://stats.stackexchange.com/questions/49243/rs-randomforest-can-not-handle-more-than-32-levels-what-is-workaround)



In [36]:

```
#=====================================================================
# Rattle timestamp: 2019-12-23 15:06:51 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# Build the train/validate/test datasets.

# nobs=1692 train=1184 validate=254 test=254

set.seed(88)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test

# The following variable selections have been noted.

crs$input     <- c("Denominator", "Year", "Source",
                   "State_Region")

crs$numeric   <- c("Denominator", "Year")

crs$categoric <- c("Source", "State_Region")

crs$target    <- "Prevalence_Risk4"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- c("State", "Prevalence", "Lower.CI", "Upper.CI", "Source_Full
crs$weights   <- NULL
```

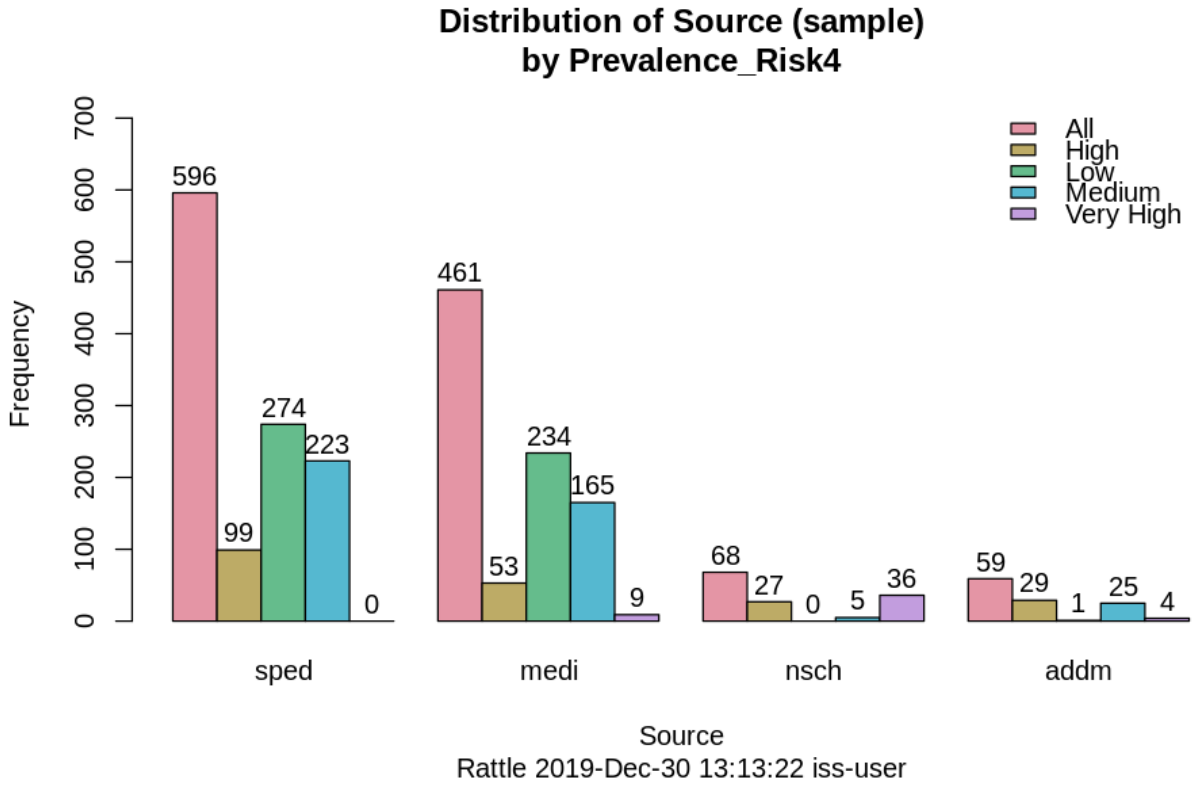
In [37]:

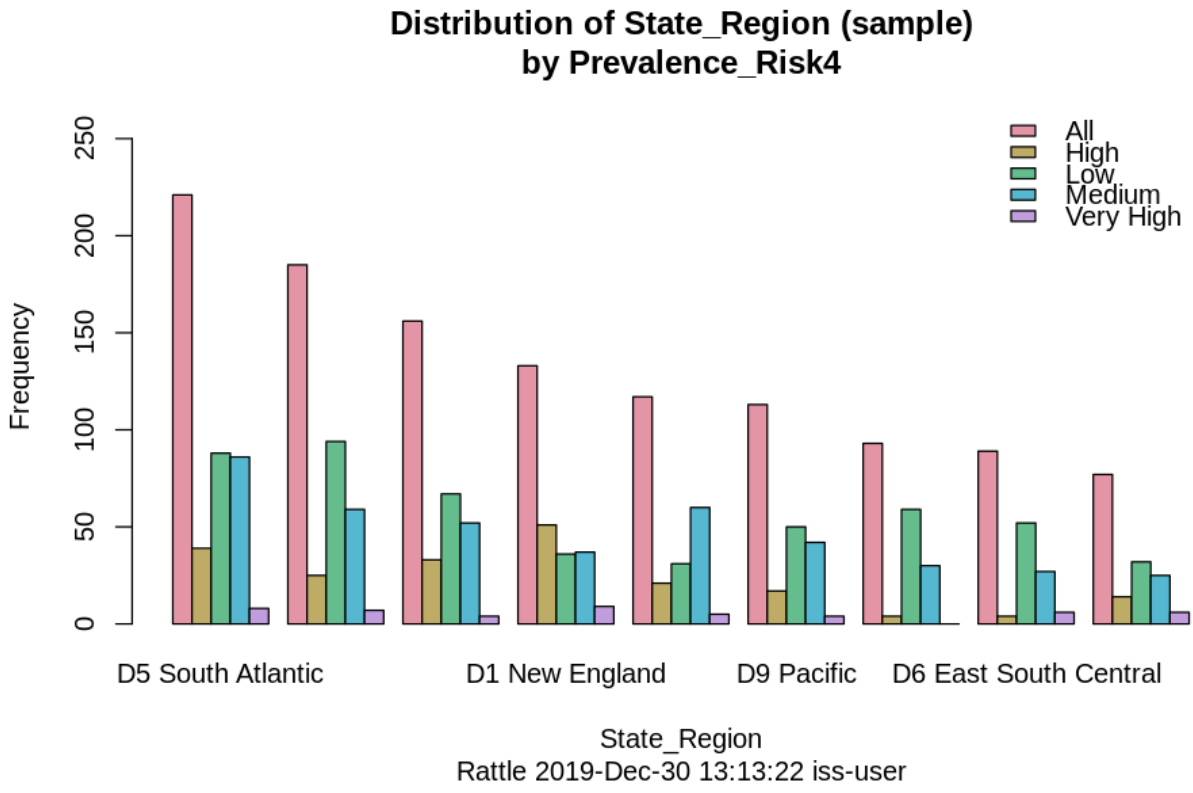
```
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=5)
```

In [38]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 14:59:20 x86_64-pc-linux-gnu

# The 'gplots' package provides the 'barplot2' function.

library(gplots, quietly=TRUE)

#============================================================================
# Rattle timestamp: 2019-12-23 14:59:20 x86_64-pc-linux-gnu

# Bar Plot

# Generate the summary data for plotting.

ds <- rbind(summary(na.omit(crs$dataset[crs$train,]$Source)),
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence

# Sort the entries.

ord <- order(ds[1,], decreasing=TRUE)

# Plot the data.

bp <-  barplot2(ds[,ord], beside=TRUE, ylab="Frequency", xlab="Source", ylim=c

# Add the actual frequencies.

text(bp, ds[,ord]+24, ds[,ord])

# Add a legend to the plot.

legend("topright", bty="n", c("All","High","Low","Medium","Very High"),  fill=

# Add a title to the plot.

title(main="Distribution of Source (sample)\nby Prevalence_Risk4",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u

#============================================================================
# Rattle timestamp: 2019-12-23 14:59:21 x86_64-pc-linux-gnu

# Bar Plot

# Generate the summary data for plotting.

ds <- rbind(summary(na.omit(crs$dataset[crs$train,]$State_Region)),
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence
    summary(na.omit(crs$dataset[crs$train,][crs$dataset[crs$train,]$Prevalence

# Sort the entries.

ord <- order(ds[1,], decreasing=TRUE)

# Plot the data.

bp <-  barplot2(ds[,ord], beside=TRUE, ylab="Frequency", xlab="State_Region",

# Add a legend to the plot.
```

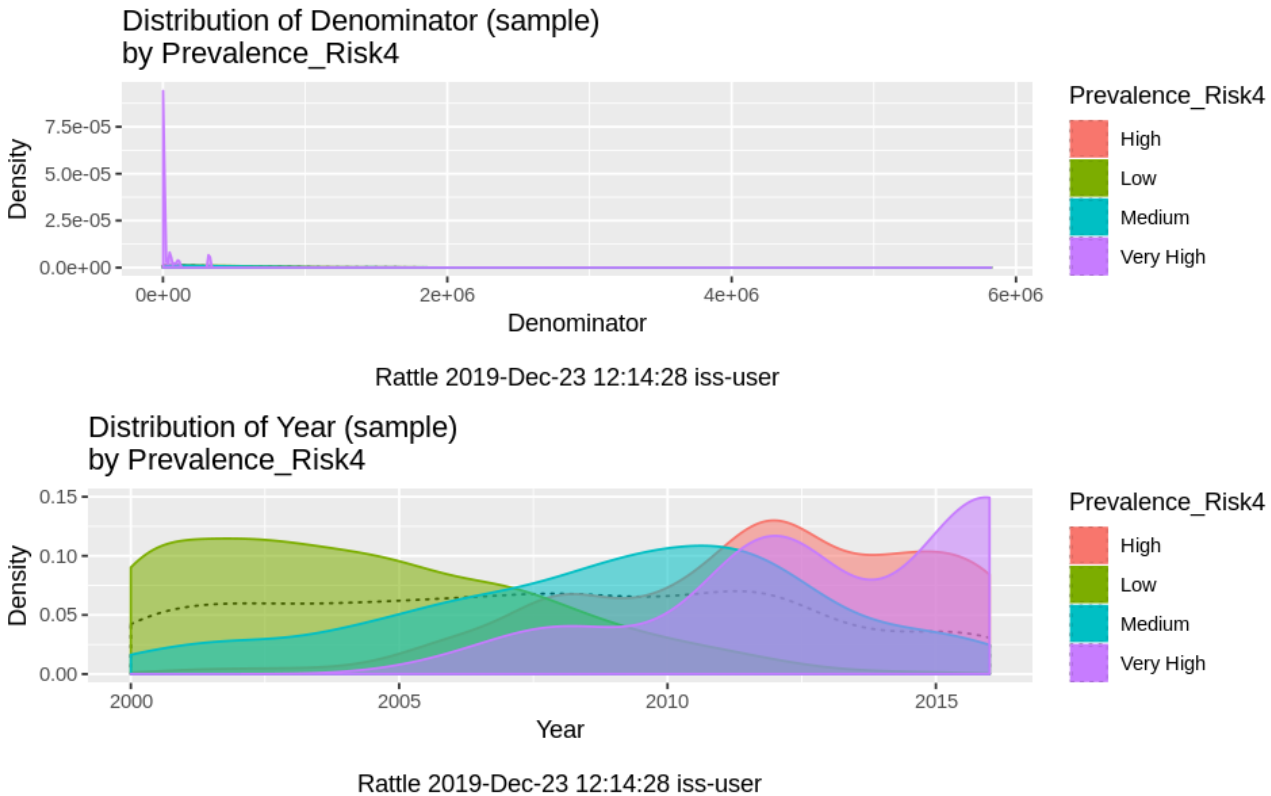
```
legend("topright", bty="n", c("All","High","Low","Medium","Very High"),  fill=

# Add a title to the plot.

title(main="Distribution of State_Region (sample)\nby Prevalence_Risk4",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
```





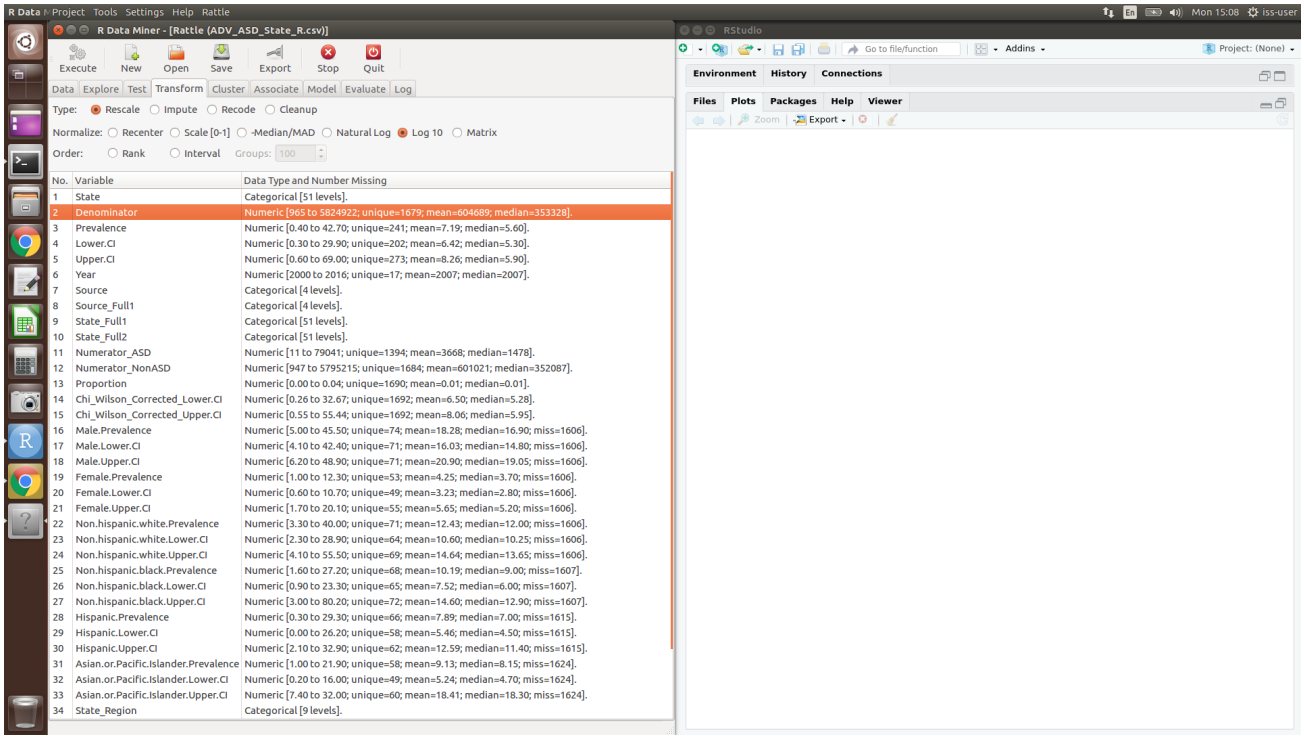
In [39]:

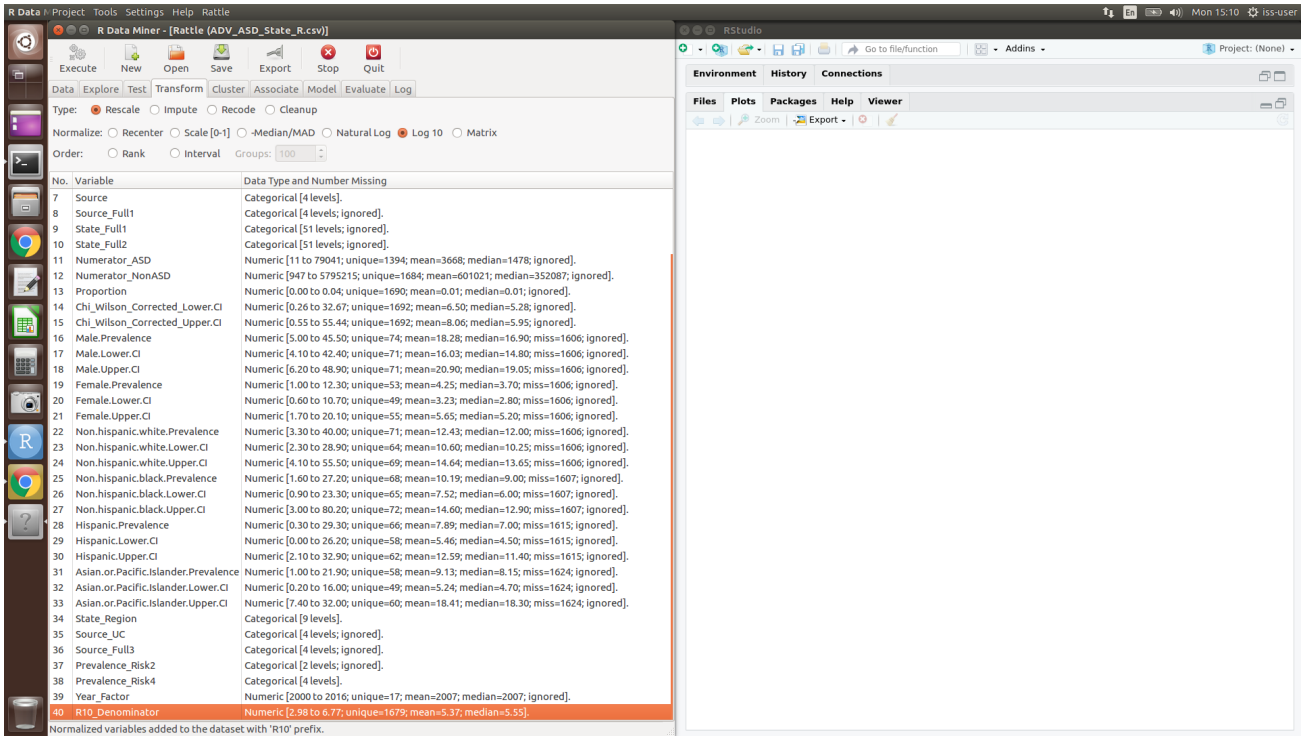
```
#=======================================================================
# Rattle timestamp: 2019-12-23 12:14:28 x86_64-pc-linux-gnu

# Display histogram plots for the selected variables.

# Use ggplot2 to generate histogram plot for Denominator

# Generate the plot.

p01 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Denominator, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Denominator)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Denominator\n\nRattle 2019-Dec-23 12:14:28 iss-user") +
  ggplot2::ggtitle("Distribution of Denominator (sample)\nby Prevalence_Risk4"
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Use ggplot2 to generate histogram plot for Year

# Generate the plot.

p02 <- crs %>%
  with(dataset[train,]) %>%
  dplyr::mutate(Prevalence_Risk4=as.factor(Prevalence_Risk4)) %>%
  dplyr::select(Year, Prevalence_Risk4) %>%
  ggplot2::ggplot(ggplot2::aes(x=Year)) +
  ggplot2::geom_density(lty=3) +
  ggplot2::geom_density(ggplot2::aes(fill=Prevalence_Risk4, colour=Prevalence_
  ggplot2::xlab("Year\n\nRattle 2019-Dec-23 12:14:28 iss-user") +
  ggplot2::ggtitle("Distribution of Year (sample)\nby Prevalence_Risk4") +
  ggplot2::labs(fill="Prevalence_Risk4", y="Density")

# Display the plots.

gridExtra::grid.arrange(p01, p02)
```



Above shows that Denominator is skrewed, which need rescale transformation

Rattle: Process & Transform Data: Rescale (log transformation)

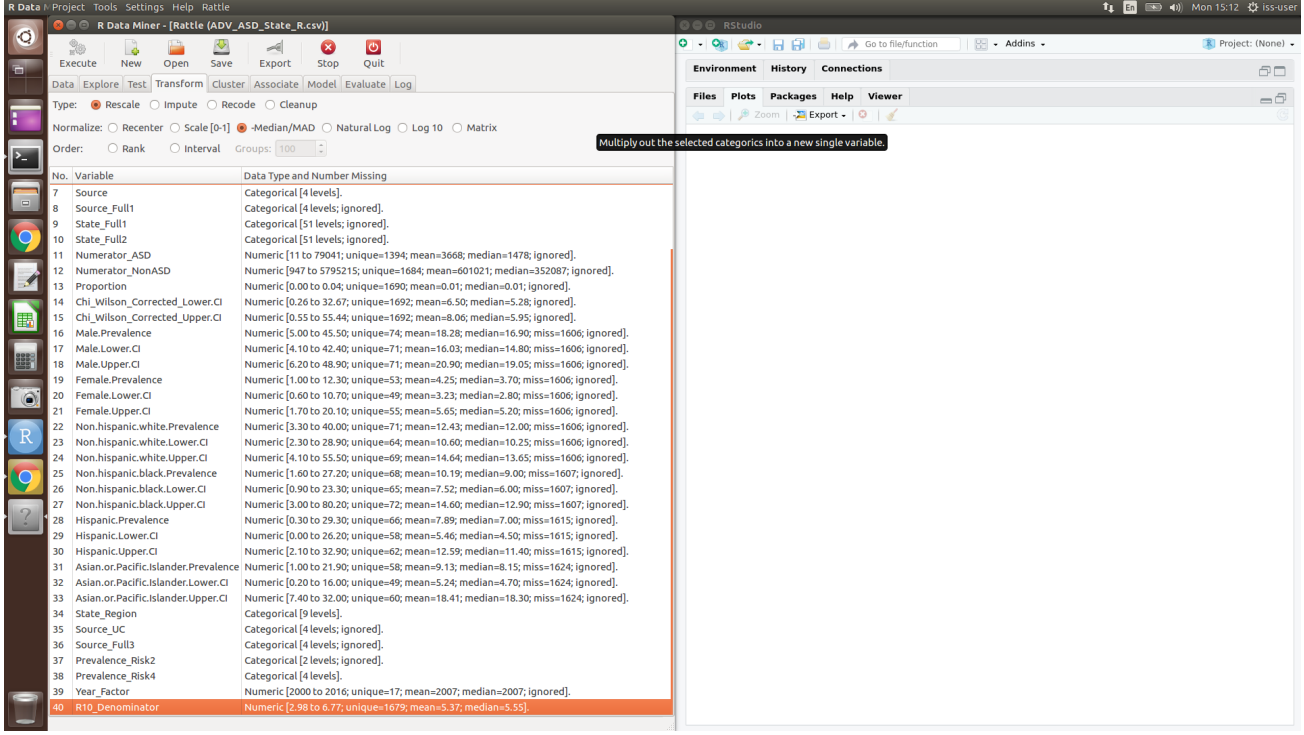


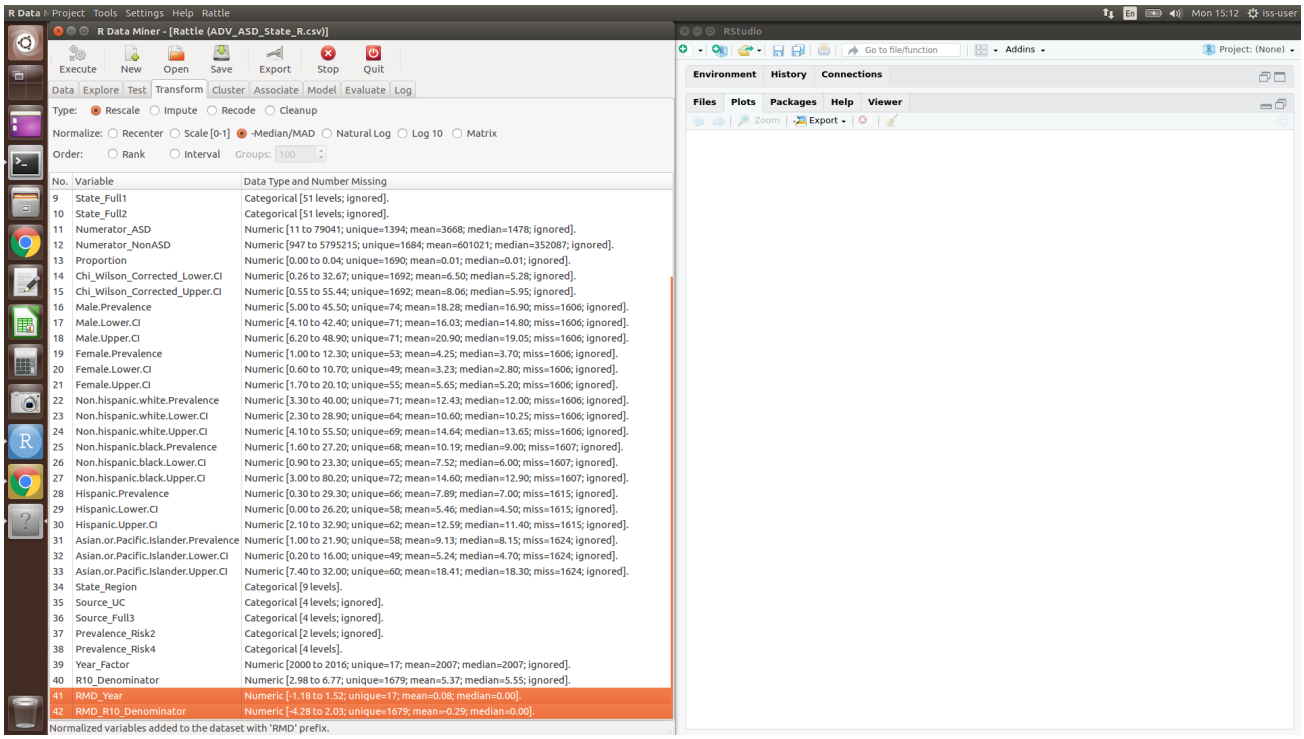


In [40]:

```
#=======================================================================
# Rattle timestamp: 2019-12-23 15:10:29 x86_64-pc-linux-gnu

# Transform variables by rescaling.

# Rescale Denominator.

crs$dataset[["R10_Denominator"]] <- crs$dataset[["Denominator"]]

# Take a log10 transform of the variable - treat -Inf as NA.

if (building)
{
  crs$dataset[["R10_Denominator"]] <-  log10(crs$dataset[["Denominator"]])
  crs$dataset[crs$dataset[["R10_Denominator"]] == -Inf & ! is.na(crs$dataset[[
}

# When scoring transform using the training data parameters.

if (scoring)
{
  crs$dataset[["R10_Denominator"]] <-  log10(crs$dataset[["Denominator"]])
  crs$dataset[crs$dataset[["R10_Denominator"]] == -Inf & ! is.na(crs$dataset[[
}

#=======================================================================
# Rattle timestamp: 2019-12-23 15:10:29 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# The following variable selections have been noted.

crs$input     <- c("Year", "Source", "State_Region",
                   "R10_Denominator")

crs$numeric   <- c("Year", "R10_Denominator")

crs$categoric <- c("Source", "State_Region")

crs$target    <- "Prevalence_Risk4"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- c("State", "Denominator", "Prevalence", "Lower.CI", "Upper.CI
crs$weights   <- NULL
```

Rattle: Process & Transform Data: Rescale (normalization)


R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Rescale Impute Recode Cleanup
Normalize: Recenter Scale [0-1] -Median/MAD Natural Log Log 10 Matrix
Order: Rank Interval Groups: 100
Multiply out the selected categorics into a new single variable.
No. Variable Data Type and Number Missing
7 Source Categorical [4 levels].
8 Source_Full1 Categorical [4 levels; ignored].
9 State_Full1 Categorical [51 levels; ignored].
10 State_Full2 Categorical [51 levels; ignored].
11 Numerator_ASD Numeric [11 to 79041; unique=1394; mean=3668; median=1478; ignored].
12 Numerator_NonASD Numeric [947 to 5795215; unique=1684; mean=601021; median=352087; ignored].
13 Proportion Numeric [0.00 to 0.04; unique=1690; mean=0.01; median=0.01; ignored].
14 Chi_Wilson_Corrected_Lower.CI Numeric [0.26 to 32.67; unique=1692; mean=6.50; median=5.28; ignored].
15 Chi_Wilson_Corrected_Upper.CI Numeric [0.55 to 55.44; unique=1692; mean=8.06; median=5.95; ignored].
16 Male.Prevalence Numeric [5.00 to 45.50; unique=74; mean=18.28; median=16.90; miss=1606; ignored].
17 Male.Lower.CI Numeric [4.10 to 42.40; unique=71; mean=16.03; median=14.80; miss=1606; ignored].
18 Male.Upper.CI Numeric [6.20 to 48.90; unique=71; mean=20.90; median=19.05; miss=1606; ignored].
19 Female.Prevalence Numeric [1.00 to 12.30; unique=53; mean=4.25; median=3.70; miss=1606; ignored].
20 Female.Lower.CI Numeric [0.60 to 10.70; unique=49; mean=3.23; median=2.80; miss=1606; ignored].
21 Female.Upper.CI Numeric [1.70 to 20.10; unique=55; mean=5.65; median=5.20; miss=1606; ignored].
22 Non.hispanic.white.Prevalence Numeric [3.30 to 40.00; unique=71; mean=12.43; median=12.00; miss=1606; ignored].
23 Non.hispanic.white.Lower.CI Numeric [2.30 to 28.90; unique=64; mean=10.60; median=10.25; miss=1606; ignored].
24 Non.hispanic.white.Upper.CI Numeric [4.10 to 55.50; unique=69; mean=14.64; median=13.65; miss=1606; ignored].
25 Non.hispanic.black.Prevalence Numeric [1.60 to 27.20; unique=68; mean=10.19; median=9.00; miss=1607; ignored].
26 Non.hispanic.black.Lower.CI Numeric [0.90 to 23.30; unique=65; mean=7.52; median=6.00; miss=1607; ignored].
27 Non.hispanic.black.Upper.CI Numeric [3.00 to 80.20; unique=72; mean=14.60; median=12.90; miss=1607; ignored].
28 Hispanic.Prevalence Numeric [0.30 to 29.30; unique=66; mean=7.89; median=7.00; miss=1615; ignored].
29 Hispanic.Lower.CI Numeric [0.00 to 26.20; unique=58; mean=5.46; median=4.50; miss=1615; ignored].
30 Hispanic.Upper.CI Numeric [2.10 to 32.90; unique=62; mean=12.59; median=11.40; miss=1615; ignored].
31 Asian.or.Pacific.Islander.Prevalence Numeric [1.00 to 21.90; unique=58; mean=9.13; median=8.15; miss=1624; ignored].
32 Asian.or.Pacific.Islander.Lower.CI Numeric [0.20 to 16.00; unique=49; mean=5.24; median=4.70; miss=1624; ignored].
33 Asian.or.Pacific.Islander.Upper.CI Numeric [7.40 to 32.00; unique=60; mean=18.41; median=18.30; miss=1624; ignored].
34 State_Region Categorical [9 levels].
35 Source_UC Categorical [4 levels; ignored].
36 Source_Full3 Categorical [4 levels; ignored].
37 Prevalence_Risk2 Categorical [2 levels; ignored].
38 Prevalence_Risk4 Categorical [4 levels].
39 Year_Factor Numeric [2000 to 2016; unique=17; mean=2007; median=2007; ignored].
40 R10_Denominator Numeric [2.98 to 6.77; unique=1679; mean=5.37; median=5.55].
RStudio
Environment History Connections
Files Plots Packages Help Viewer
Addins
Project: (None)
Export
Zoom

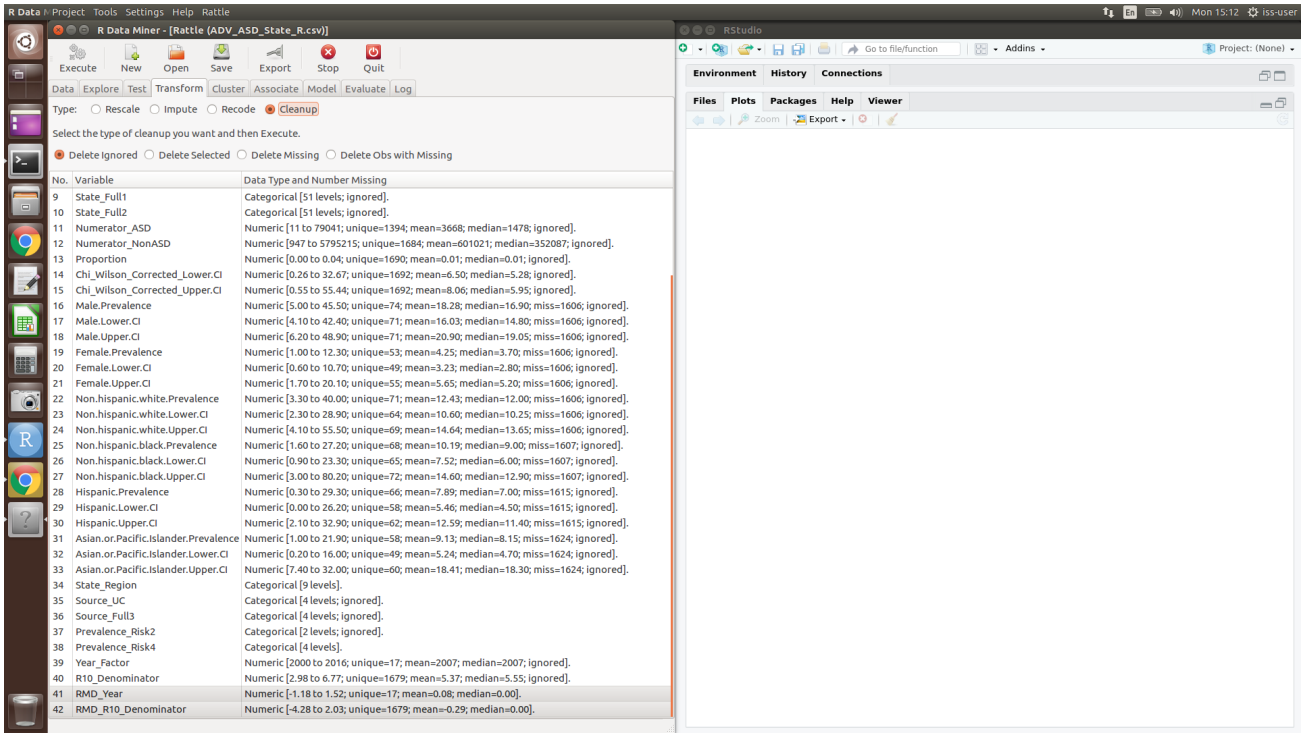

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Rescale Impute Recode Cleanup
Normalize: Recenter Scale [0-1] -Median/MAD Natural Log Log 10 Matrix
Order: Rank Interval Groups: 100
No. Variable Data Type and Number Missing
9 State_Full1 Categorical [51 levels; ignored].
10 State_Full2 Categorical [51 levels; ignored].
11 Numerator_ASD Numeric [11 to 79041; unique=1394; mean=3668; median=1478; ignored].
12 Numerator_NonASD Numeric [947 to 5795215; unique=1684; mean=601021; median=352087; ignored].
13 Proportion Numeric [0.00 to 0.04; unique=1690; mean=0.01; median=0.01; ignored].
14 Chi_Wilson_Corrected_Lower.CI Numeric [0.26 to 32.67; unique=1692; mean=6.50; median=5.28; ignored].
15 Chi_Wilson_Corrected_Upper.CI Numeric [0.55 to 55.44; unique=1692; mean=8.06; median=5.95; ignored].
16 Male.Prevalence Numeric [5.00 to 45.50; unique=74; mean=18.28; median=16.90; miss=1606; ignored].
17 Male.Lower.CI Numeric [4.10 to 42.40; unique=71; mean=16.03; median=14.80; miss=1606; ignored].
18 Male.Upper.CI Numeric [6.20 to 48.90; unique=71; mean=20.90; median=19.05; miss=1606; ignored].
19 Female.Prevalence Numeric [1.00 to 12.30; unique=53; mean=4.25; median=3.70; miss=1606; ignored].
20 Female.Lower.CI Numeric [0.60 to 10.70; unique=49; mean=3.23; median=2.80; miss=1606; ignored].
21 Female.Upper.CI Numeric [1.70 to 20.10; unique=55; mean=5.65; median=5.20; miss=1606; ignored].
22 Non.hispanic.white.Prevalence Numeric [3.30 to 40.00; unique=71; mean=12.43; median=12.00; miss=1606; ignored].
23 Non.hispanic.white.Lower.CI Numeric [2.30 to 28.90; unique=64; mean=10.60; median=10.25; miss=1606; ignored].
24 Non.hispanic.white.Upper.CI Numeric [4.10 to 55.50; unique=69; mean=14.64; median=13.65; miss=1606; ignored].
25 Non.hispanic.black.Prevalence Numeric [1.60 to 27.20; unique=68; mean=10.19; median=9.00; miss=1607; ignored].
26 Non.hispanic.black.Lower.CI Numeric [0.90 to 23.30; unique=65; mean=7.52; median=6.00; miss=1607; ignored].
27 Non.hispanic.black.Upper.CI Numeric [3.00 to 80.20; unique=72; mean=14.60; median=12.90; miss=1607; ignored].
28 Hispanic.Prevalence Numeric [0.30 to 29.30; unique=66; mean=7.89; median=7.00; miss=1615; ignored].
29 Hispanic.Lower.CI Numeric [0.00 to 26.20; unique=58; mean=5.46; median=4.50; miss=1615; ignored].
30 Hispanic.Upper.CI Numeric [2.10 to 32.90; unique=62; mean=12.59; median=11.40; miss=1615; ignored].
31 Asian.or.Pacific.Islander.Prevalence Numeric [1.00 to 21.90; unique=58; mean=9.13; median=8.15; miss=1624; ignored].
32 Asian.or.Pacific.Islander.Lower.CI Numeric [0.20 to 16.00; unique=49; mean=5.24; median=4.70; miss=1624; ignored].
33 Asian.or.Pacific.Islander.Upper.CI Numeric [7.40 to 32.00; unique=60; mean=18.41; median=18.30; miss=1624; ignored].
34 State_Region Categorical [9 levels].
35 Source_UC Categorical [4 levels; ignored].
36 Source_Full3 Categorical [4 levels; ignored].
37 Prevalence_Risk2 Categorical [2 levels; ignored].
38 Prevalence_Risk4 Categorical [4 levels].
39 Year_Factor Numeric [2000 to 2016; unique=17; mean=2007; median=2007; ignored].
40 R10_Denominator Numeric [2.98 to 6.77; unique=1679; mean=5.37; median=5.55; ignored].
41 RMD_Year Numeric [-1.18 to 1.52; unique=17; mean=0.08; median=0.00].
42 RMD_R10_Denominator Numeric [-4.28 to 2.03; unique=1679; mean=-0.29; median=0.00].
Normalized variables added to the dataset with 'RMD' prefix.
RStudio
Environment History Connections
Files Plots Packages Help Viewer
Addins
Project: (None)
Export
Zoom

In [41]:

```
#=====================================================================
# Rattle timestamp: 2019-12-23 15:12:22 x86_64-pc-linux-gnu

# Transform variables by rescaling.

# The 'reshape' package provides the 'rescaler' function.

library(reshape, quietly=TRUE)

# Rescale Year.

crs$dataset[["RMD_Year"]] <- crs$dataset[["Year"]]

# Rescale by subtracting median and dividing by median abs deviation.

if (building)
{
  crs$dataset[["RMD_Year"]] <-  rescaler(crs$dataset[["Year"]], "robust")
}

# When scoring transform using the training data parameters.

if (scoring)
{
  crs$dataset[["RMD_Year"]] <- (crs$dataset[["Year"]] - 2007.000000)/5.930400
}

# Rescale R10_Denominator.

crs$dataset[["RMD_R10_Denominator"]] <- crs$dataset[["R10_Denominator"]]

# Rescale by subtracting median and dividing by median abs deviation.

if (building)
{
  crs$dataset[["RMD_R10_Denominator"]] <-  rescaler(crs$dataset[["R10_Denomina
}

# When scoring transform using the training data parameters.

if (scoring)
{
  crs$dataset[["RMD_R10_Denominator"]] <- (crs$dataset[["R10_Denominator"]] -
}

#=====================================================================
# Rattle timestamp: 2019-12-23 15:12:23 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# The following variable selections have been noted.

crs$input     <- c("Source", "State_Region", "RMD_Year",
                   "RMD_R10_Denominator")

crs$numeric   <- c("RMD_Year", "RMD_R10_Denominator")

crs$categoric <- c("Source", "State_Region")

crs$target    <- "Prevalence_Risk4"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- c("State", "Denominator", "Prevalence", "Lower.CI", "Upper.CI
```

```
crs$weights    <- NULL
```

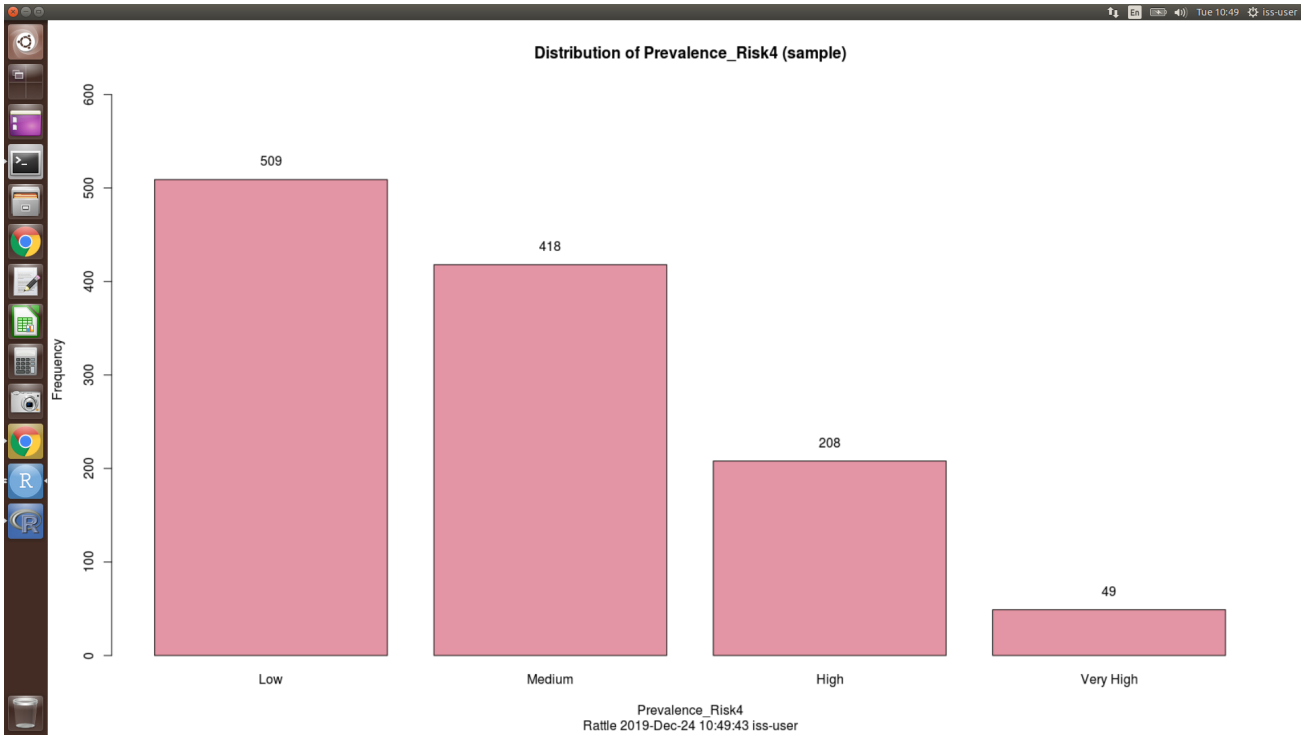
Optionally, Cleanup variables by Delete Ignored



()

Rattle: Train Model (Classification)

Multi-Class Classification: Prevalence_Risk4



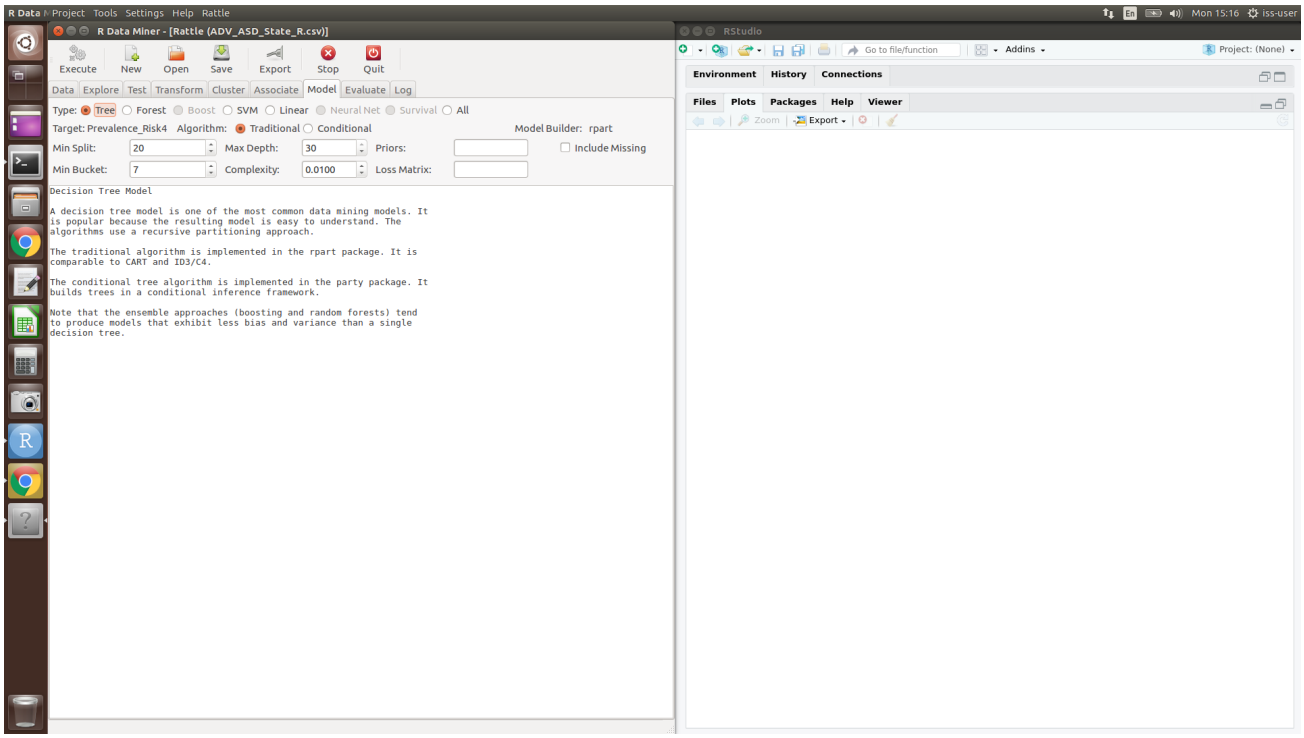
In [42]:
```
crs$target
```

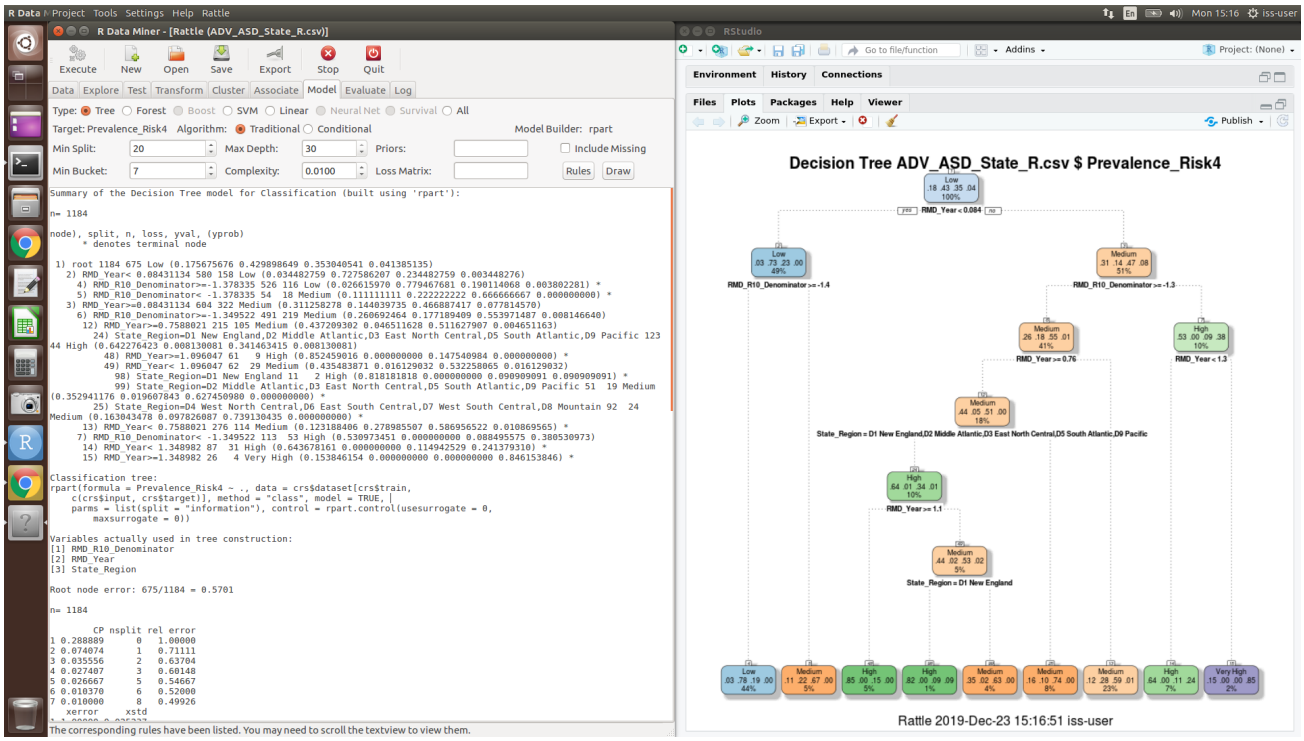
'Prevalence_Risk4'

In [43]:
```
crs$input
```

'Source' 'State_Region' 'RMD_Year' 'RMD_R10_Denominator'

Multi-Class Model: Decision Tree (DT)





In [44]:
```
if(!require(rpart)){install.packages("rpart")}
library('rpart')
```

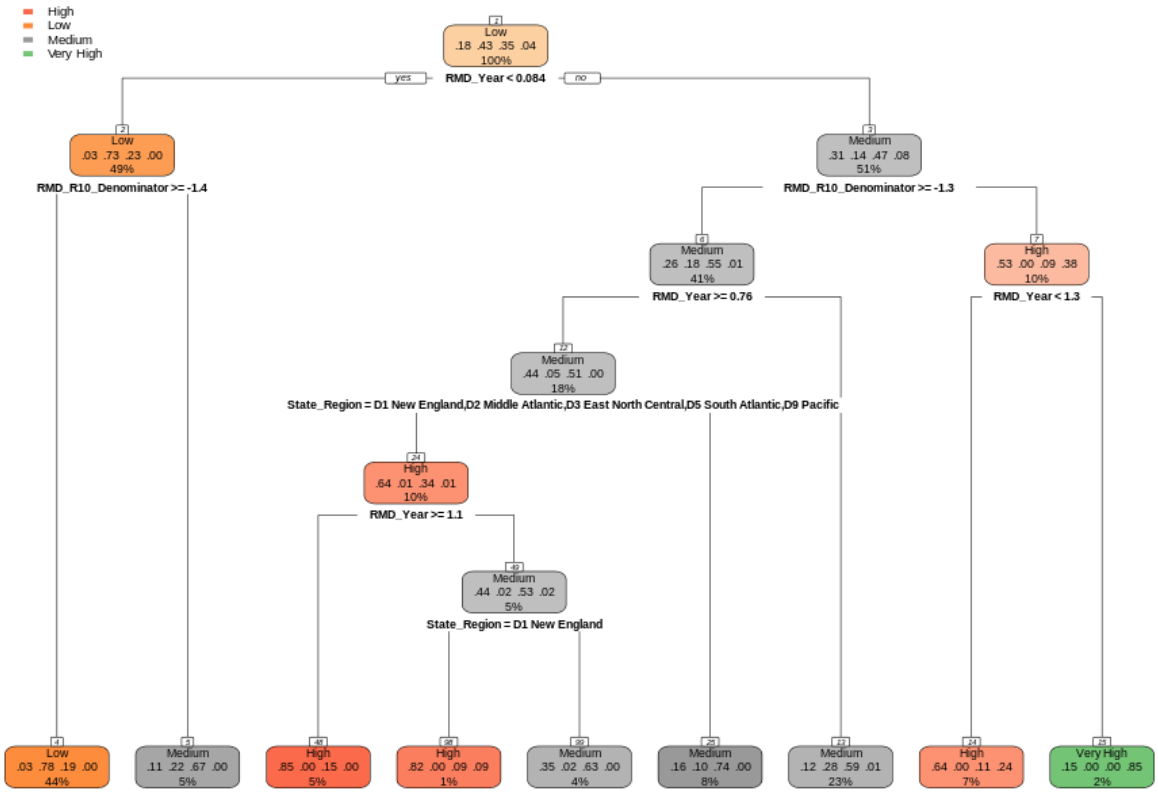
```
Loading required package: rpart
```

In [45]:
```
#=============================================================================
# Rattle timestamp: 2019-12-23 15:16:48 x86_64-pc-linux-gnu

# Decision Tree

# The 'rpart' package provides the 'rpart' function.

library(rpart, quietly=TRUE)

# Reset the random number seed to obtain the same results each time.

set.seed(crv$seed)

# Build the Decision Tree model.

crs$rpart <- rpart(Prevalence_Risk4 ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="class",
    parms=list(split="information"),
    control=rpart.control(usesurrogate=0,
        maxsurrogate=0),
    model=TRUE)

# Generate a textual view of the Decision Tree model.

print(crs$rpart)
printcp(crs$rpart)
cat("\n")

# Time taken: 0.05 secs

# List the rules from the tree using a Rattle support function.

asRules(crs$rpart)
```

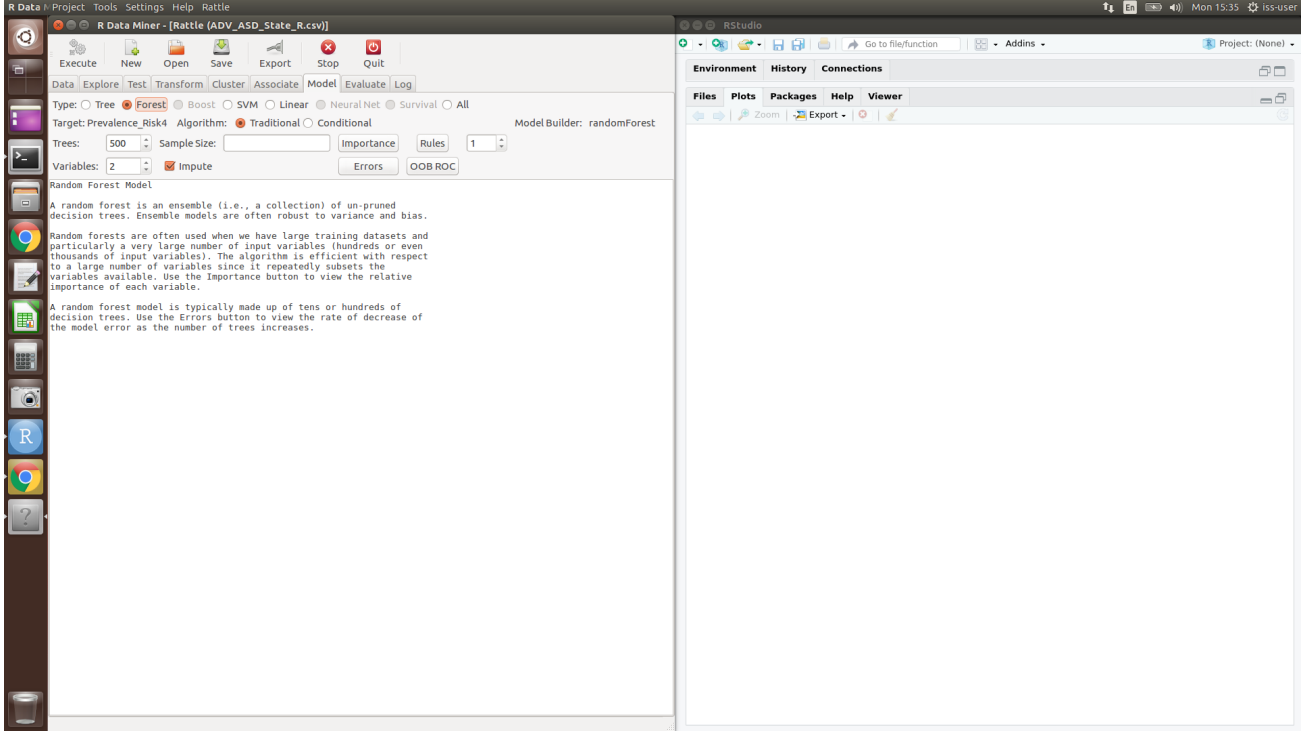
```
n= 1184

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1184 675 Low (0.175675676 0.429898649 0.353040541 0.041385135)
   2) RMD_Year< 0.08431134 580 158 Low (0.034482759 0.727586207 0.234482759
0.003448276)
     4) RMD_R10_Denominator>=-1.378335 526 116 Low (0.026615970 0.779467681
0.190114068 0.003802281) *
     5) RMD_R10_Denominator< -1.378335 54  18 Medium (0.111111111 0.2222222
22 0.666666667 0.000000000) *
   3) RMD_Year>=0.08431134 604 322 Medium (0.311258278 0.144039735 0.466887
417 0.077814570)
     6) RMD_R10_Denominator>=-1.349522 491 219 Medium (0.260692464 0.177189
409 0.553971487 0.008146640)
      12) RMD_Year>=0.7588021 215 105 Medium (0.437209302 0.046511628 0.511
627907 0.004651163)
        24) State_Region=D1 New England,D2 Middle Atlantic,D3 East North Ce
ntral,D5 South Atlantic,D9 Pacific 123  44 High (0.642276423 0.008130081 0
```

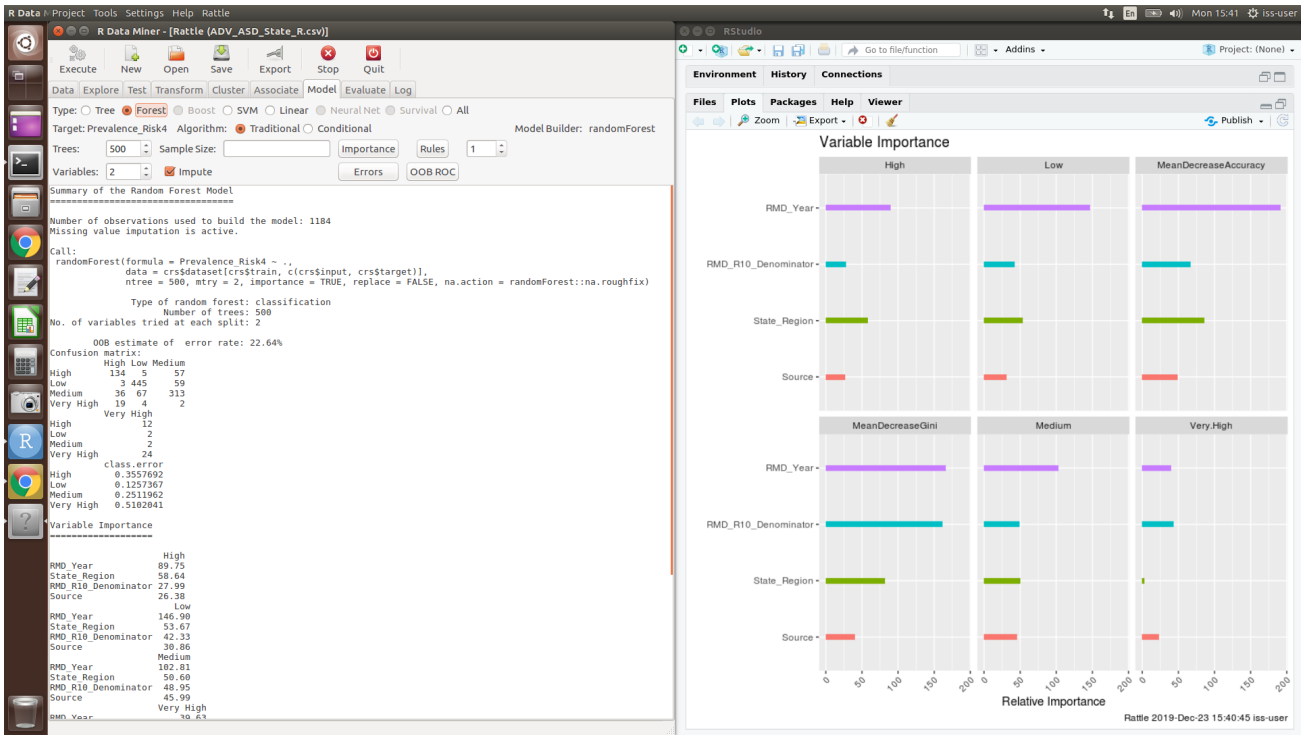
In [46]:
```
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=6)
```

In [47]:
```
rpart.plot::prp(crs$rpart,
    type = 2, extra = "auto", nn = TRUE,
    under = FALSE, fallen.leaves = TRUE,
    digits = 2, varlen = 0, faclen = 0,
#    roundint = TRUE,
    cex = NULL, tweak = 1,
#    clip.facs = FALSE,
    clip.right.labs = TRUE,
    snip = FALSE,
    box.palette = "auto", shadow.col = 0)
```



Multi-Class Model: Random Forest (RF)


R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Target: Prevalence_Risk4 Algorithm: Traditional Conditional Model Builder: randomForest
Trees: 500 Sample Size: Importance Rules 1
Variables: 2 Impute Errors OOB ROC
Random Forest Model
A random forest is an ensemble (i.e., a collection) of un-pruned decision trees. Ensemble models are often robust to variance and bias.
Random forests are often used when we have large training datasets and particularly a very large number of input variables (hundreds or even thousands of input variables). The algorithm is efficient with respect to a large number of variables since it repeatedly subsets the variables available. Use the Importance button to view the relative importance of each variable.
A random forest model is typically made up of tens or hundreds of decision trees. Use the Errors button to view the rate of decrease of the model error as the number of trees increases.
RStudio
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom
Export

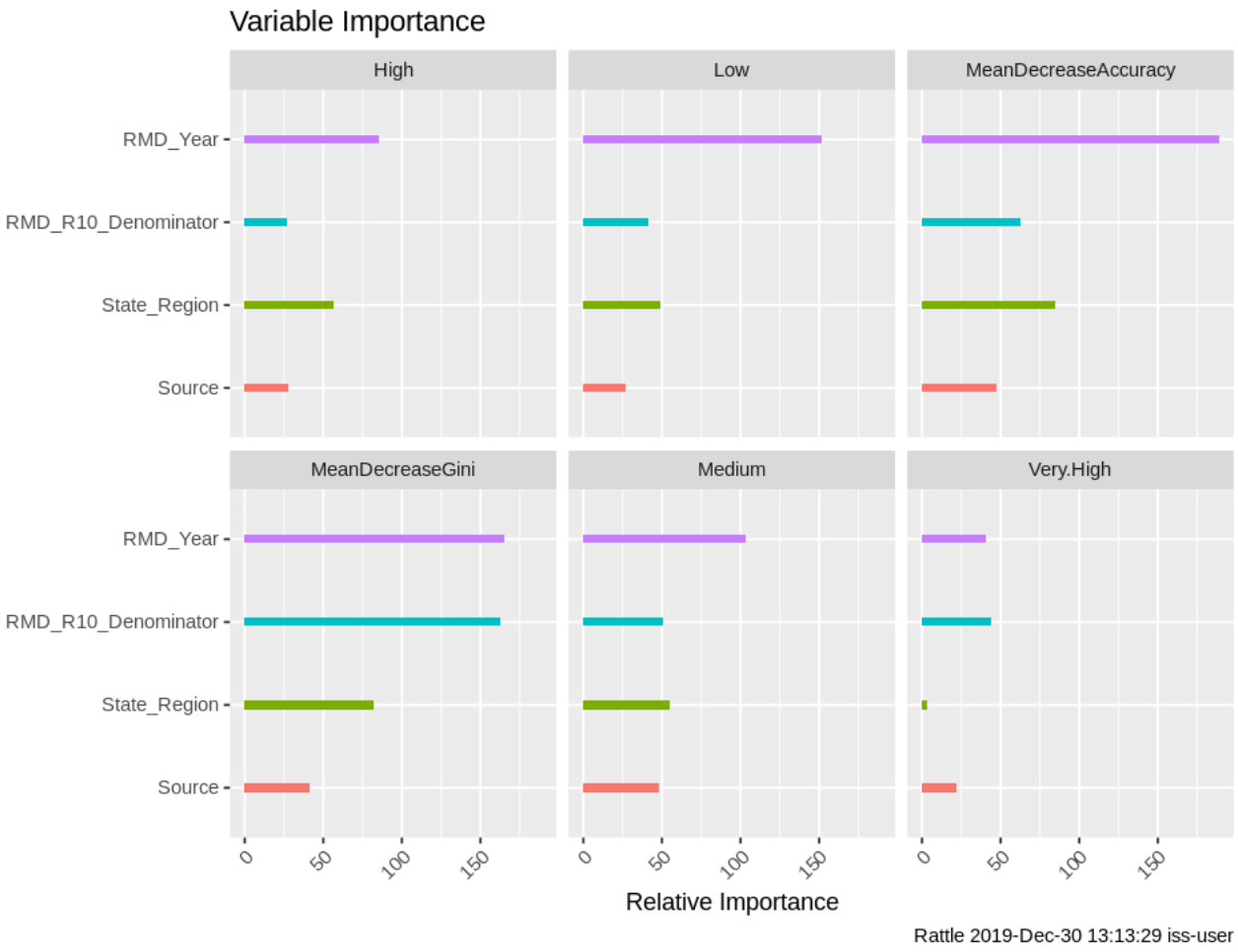

R Data Miner - [Rattle (ADV_ASD_State_R.csv)]
Execute
New
Open
Save
Export
Stop
Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Target: Prevalence_Risk4 Algorithm: Traditional Conditional Model Builder: randomForest
Trees: 500 Sample Size: Importance Rules 1
Variables: 2 Impute Errors OOB ROC
Summary of the Random Forest Model
Number of observations used to build the model: 1184
Missing value imputation is active.
Call:
randomForest(formula = Prevalence_Risk4 ~ ., data = crs$dataset[crs$train, c(crs$input, crs$target)], ntree = 500, mtry = 2, importance = TRUE, replace = FALSE, na.action = randomForest::na.roughfix)
Type of random forest: classification
Number of trees: 500
No. of variables tried at each split: 2
OOB estimate of error rate: 22.64%
Confusion matrix:
High Low Medium
High 134 5 57
Low 3 445 59
Medium 36 67 313
Very High 19 4 2
Very High
High 12
Low 2
Medium 2
Very High 24
class.error
High 0.3557692
Low 0.1257367
Medium 0.2511962
Very High 0.5102041
Variable Importance
High
RMD_Year 89.75
State_Region 58.64
RMD_R10_Denominator 27.99
Source 26.38
Low
RMD_Year 146.90
State_Region 53.67
RMD_R10_Denominator 42.33
Source 30.86
Medium
RMD_Year 102.81
State_Region 50.60
RMD_R10_Denominator 48.95
Source 45.99
Very High
RMD_Year 39.63
RStudio
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom
Export
Publish
Variable Importance
High
Low
MeanDecreaseAccuracy
RMD_Year
RMD_R10_Denominator
State_Region
Source
MeanDecreaseGini
Medium
Very.High
0 50 100 150 200
Relative Importance
Rattle 2019-Dec-23 15:40:45 iss-user

In [48]:

```
#=====================================================================
# Rattle timestamp: 2019-12-23 15:40:42 x86_64-pc-linux-gnu

# Build a Random Forest model using the traditional approach.

set.seed(crv$seed)

crs$rf <- randomForest::randomForest(Prevalence_Risk4 ~ .,
  data=crs$dataset[crs$train, c(crs$input, crs$target)],
  ntree=500,
  mtry=2,
  importance=TRUE,
  na.action=randomForest::na.roughfix,
  replace=FALSE)

# Generate textual output of the 'Random Forest' model.

crs$rf

# List the importance of the variables.

rn <- round(randomForest::importance(crs$rf), 2)
rn[order(rn[,3], decreasing=TRUE),]

# Time taken: 2.00 secs

#=====================================================================
# Rattle timestamp: 2019-12-23 15:40:45 x86_64-pc-linux-gnu

# Plot the relative importance of the variables.

p <- ggVarImp(crs$rf,
              title="Variable Importance Random Forest ADV_ASD_State_R.csv")
p
```

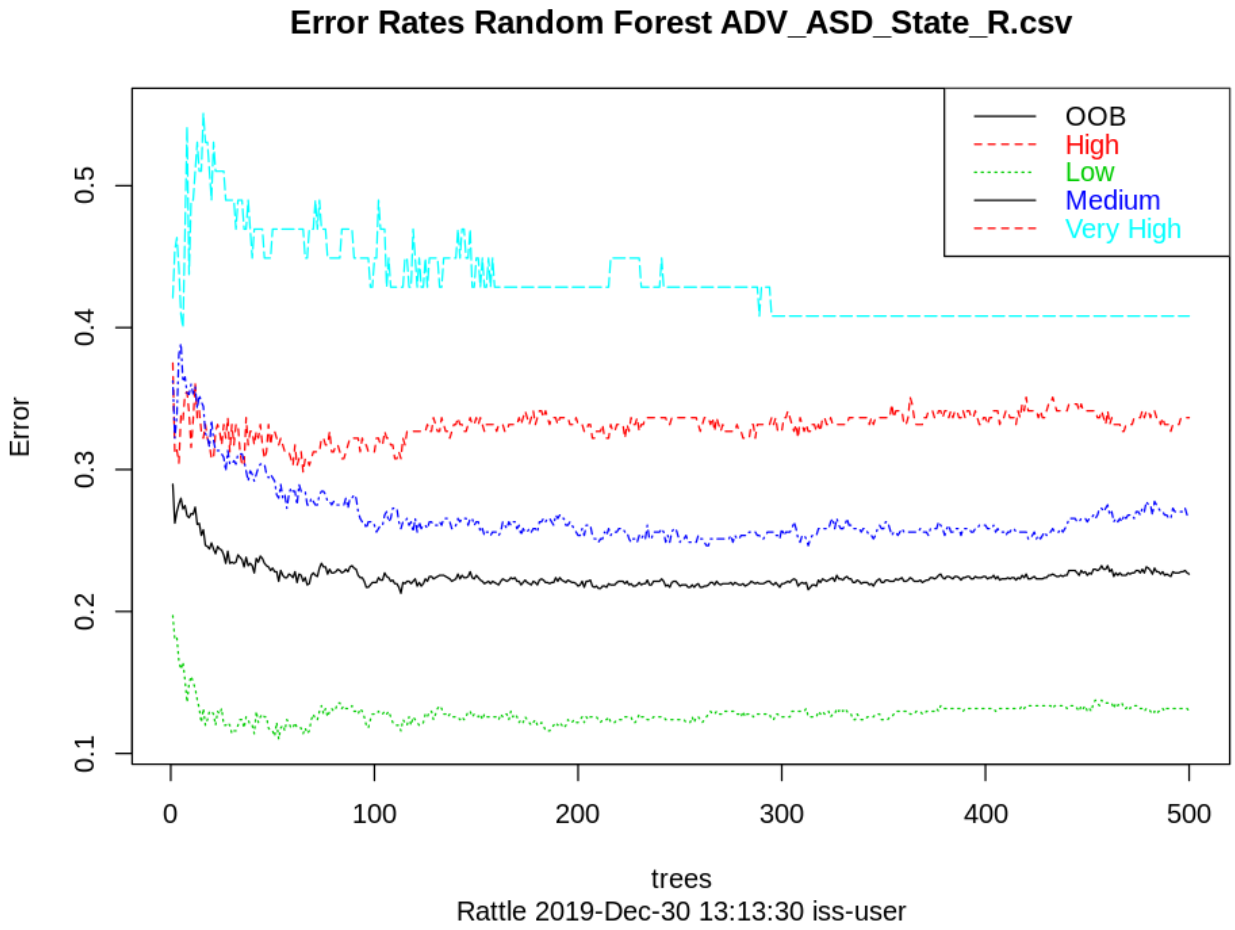
```
Call:
 randomForest(formula = Prevalence_Risk4 ~ ., data = crs$dataset[crs$train,
    c(crs$input, crs$target)], ntree = 500, mtry = 2, importance = TRUE,
replace = FALSE, na.action = randomForest::na.roughfix)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 22.64%
Confusion matrix:
          High Low Medium Very High class.error
High       138   4     55        11   0.3365385
Low          3 443     60         3   0.1296660
Medium      40  70    306         2   0.2679426
Very High   14   4      2        29   0.4081633
```

	High	Low	Medium	Very High	MeanDecreaseAccuracy	MeanDecreaseGini
RMD_Year	85.07	151.93	103.16	40.59	188.83	164.85
State_Region	56.22	49.25	54.81	3.18	84.47	81.96
RMD_R10_Denominator	26.47	41.32	50.31	43.59	62.56	162.78
Source	28.10	27.31	48.10	21.90	47.57	41.03


Variable Importance
High
Low
MeanDecreaseAccuracy
MeanDecreaseGini
Medium
Very.High
RMD_Year
RMD_R10_Denominator
State_Region
Source
0
50
100
150
Relative Importance
Rattle 2019-Dec-30 13:13:29 iss-user

In [49]:

```
# Plot the error rate against the number of trees.

plot(crs$rf, main="")
legend("topright", c("OOB", "High", "Low", "Medium", "Very High"), text.col=1:
title(main="Error Rates Random Forest ADV_ASD_State_R.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
```



In [50]:

```
if(!require(verification)){install.packages("verification")}
library('verification')
```

```
Loading required package: verification
Loading required package: fields
Loading required package: spam
Loading required package: dotCall64
Loading required package: grid
Spam version 2.5-1 (2019-12-12) is loaded.
Type 'help( Spam)' or 'demo( spam)' for a short introduction
and overview of this package.
Help for individual functions is also obtained by adding the
suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

Attaching package: ‘spam’

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

    backsolve, forwardsolve

Loading required package: maps
See https://github.com/NCAR/Fields (https://github.com/NCAR/Fields) for
```

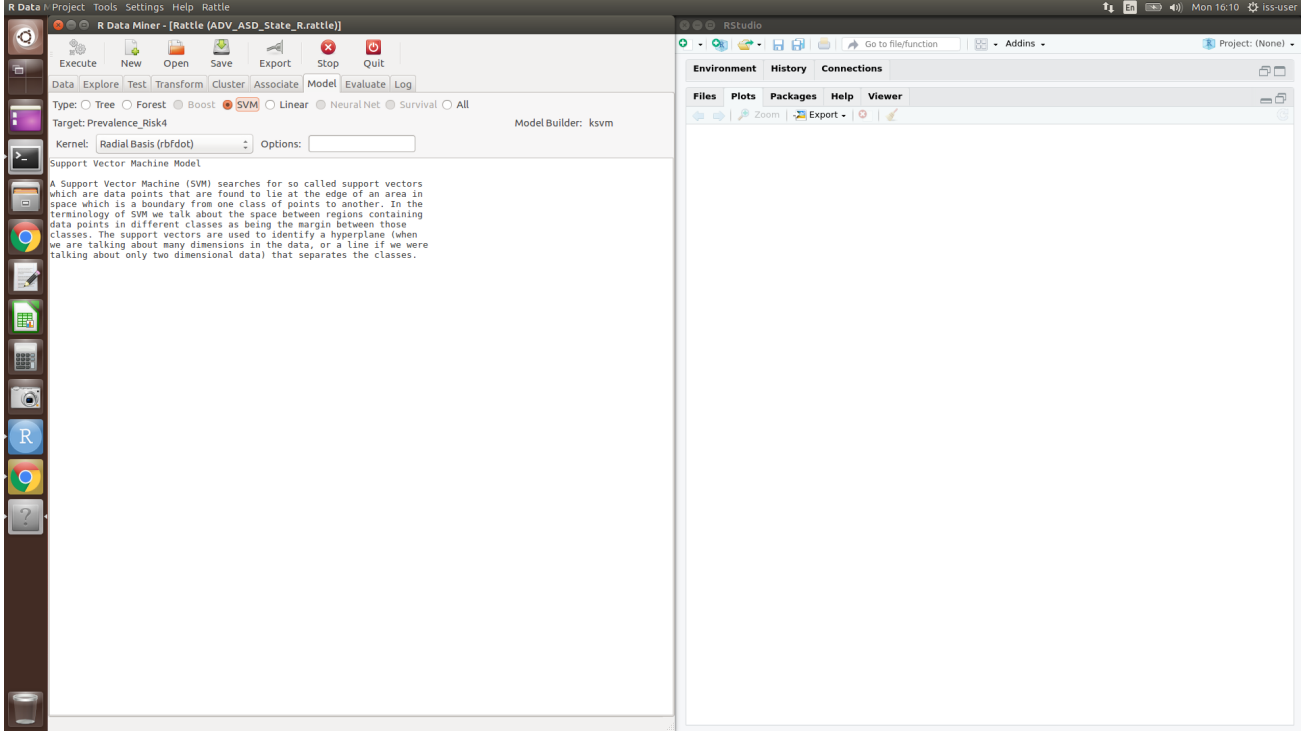
In [51]:
```
head(crs$rf$votes)
```

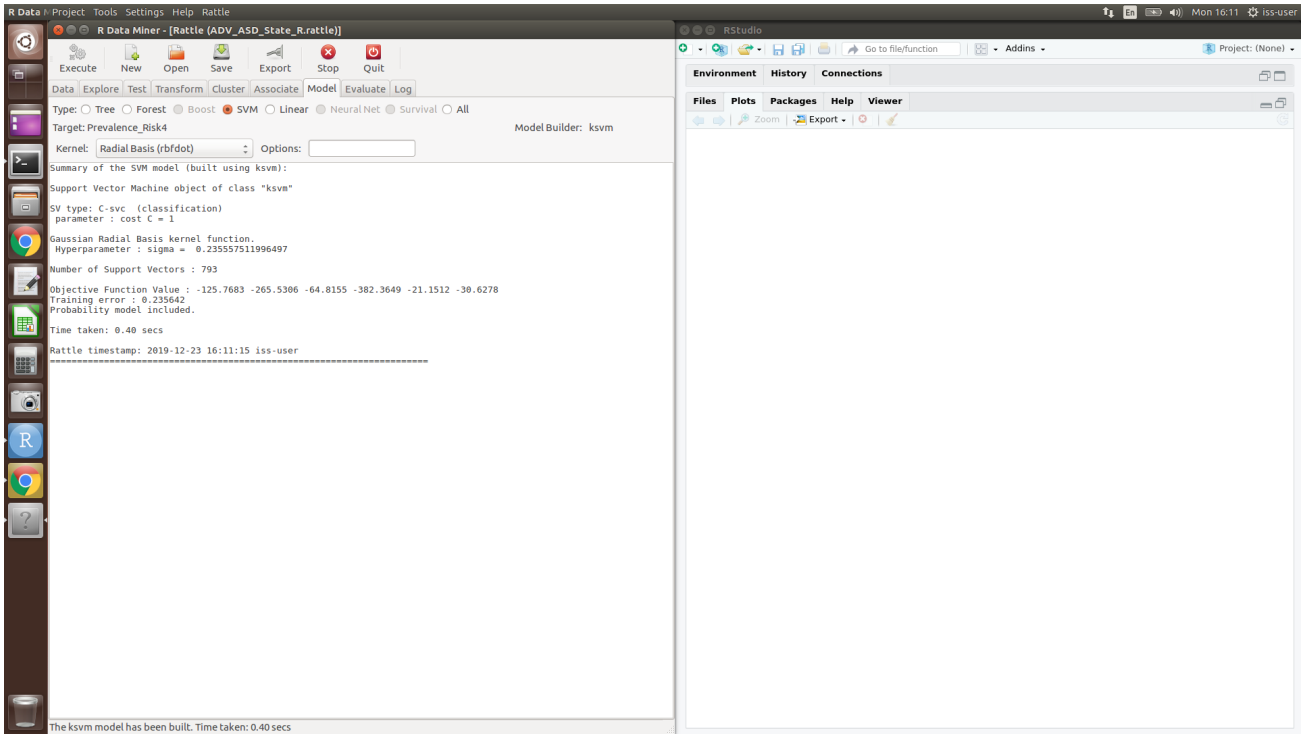
	High	Low	Medium	Very High
279	0.00000000	0.98870056	0.01129944	0.0000000
589	0.32820513	0.26153846	0.41025641	0.0000000
1462	0.24309392	0.00000000	0.75690608	0.0000000
743	0.02824859	0.67231638	0.29943503	0.0000000
1427	0.62637363	0.02197802	0.35164835	0.0000000
81	0.57627119	0.00000000	0.11299435	0.3107345

In [52]:
```
# Display tree number 1 to 3.

printRandomForests(crs$rf, 1:3)
```

```
Random Forest Model 1

-----------------------------------------------------------------
Tree 1 Rule 1 Node 16 Decision Very High

1: RMD_R10_Denominator <= -3.86457272986912
2: RMD_Year <= 0.505868069607446
3: State_Region IN ("D3 East North Central")
4: RMD_R10_Denominator <= -3.93592015489017
-----------------------------------------------------------------
Tree 1 Rule 2 Node 17 Decision High

1: RMD_R10_Denominator <= -3.86457272986912
2: RMD_Year <= 0.505868069607446
3: State_Region IN ("D3 East North Central")
4: RMD_R10_Denominator > -3.93592015489017
-----------------------------------------------------------------
Tree 1 Rule 3 Node 9 Decision High

1: RMD_R10_Denominator <= -3.86457272986912
```

Multi-Class Model: Support Vector Machines (SVM)





In [53]:
```
if(!require(kernlab)){install.packages("kernlab")}
library('kernlab')
```

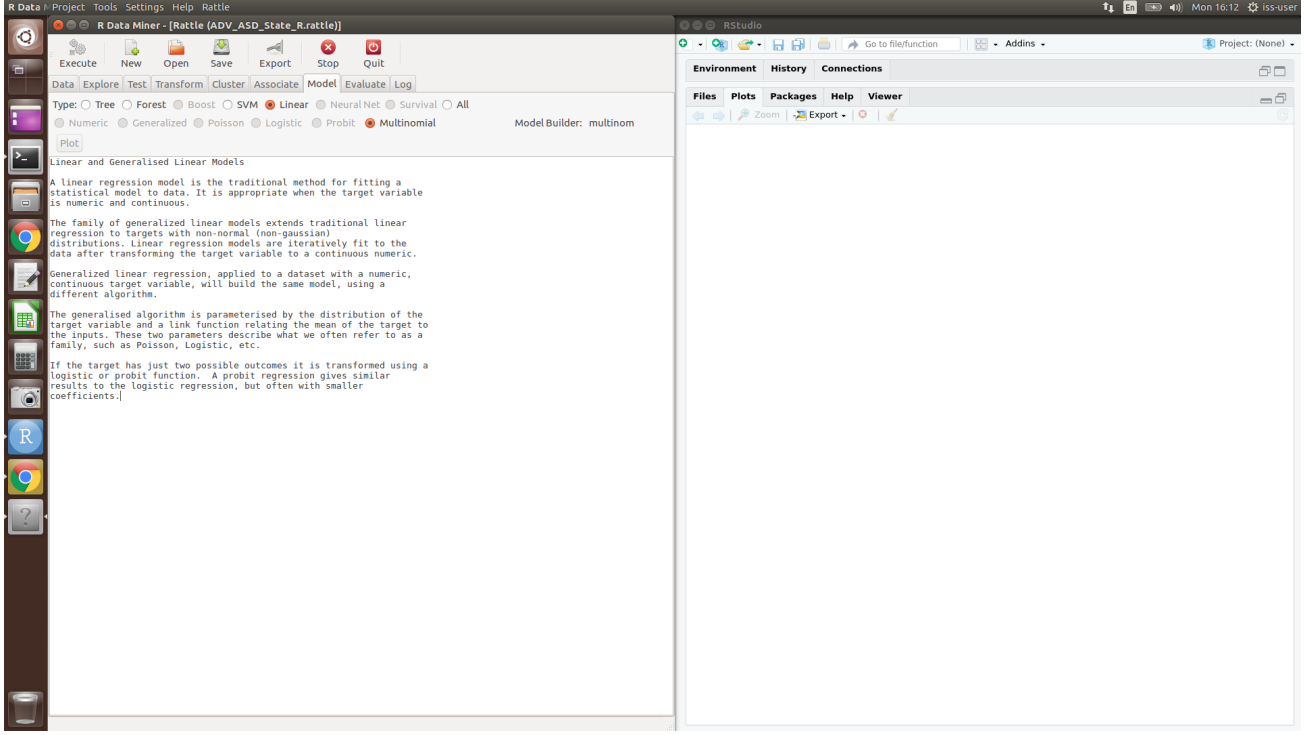
```
Loading required package: kernlab

Attaching package: ‘kernlab’

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

    rvm

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

    alpha
```

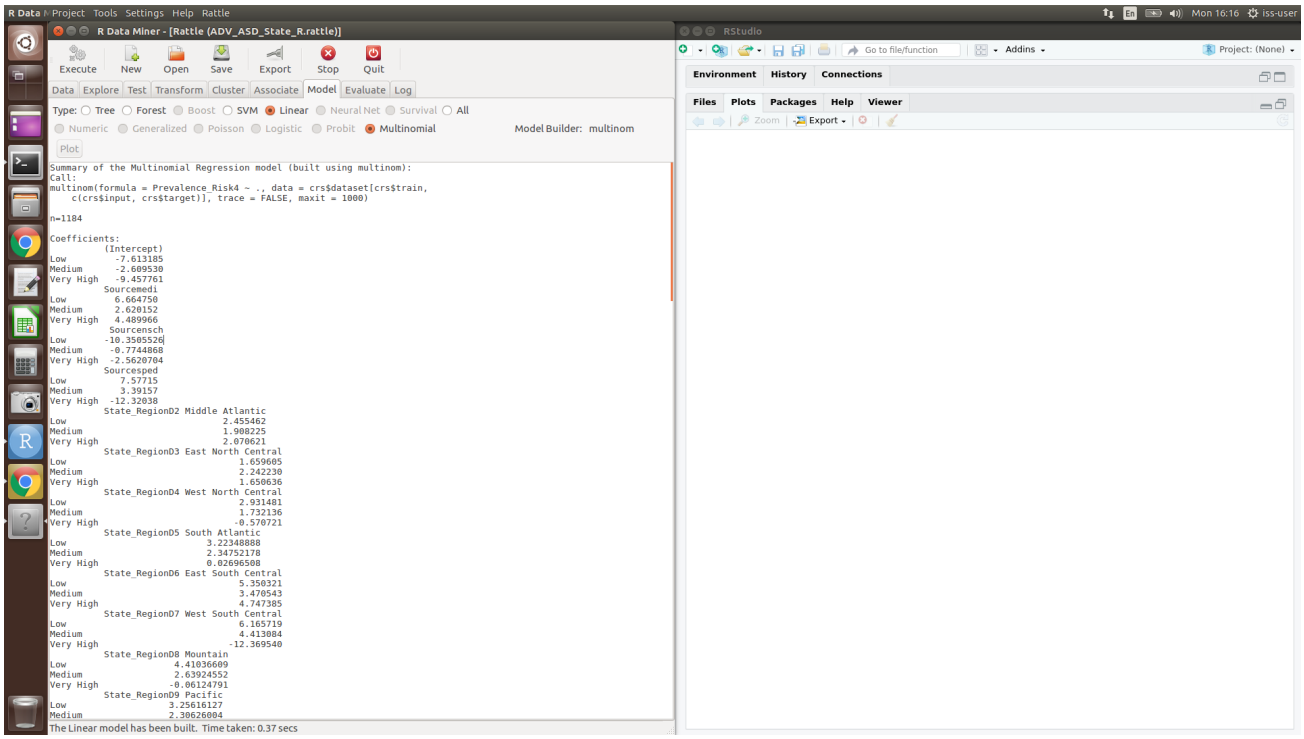
In [54]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 16:17:34 x86_64-pc-linux-gnu

# Support vector machine.

# The 'kernlab' package provides the 'ksvm' function.

library(kernlab, quietly=TRUE)

# Build a Support Vector Machine model.

set.seed(crv$seed)
crs$ksvm <- ksvm(as.factor(Prevalence_Risk4) ~ .,
      data=crs$dataset[crs$train,c(crs$input, crs$target)],
      kernel="rbfdot",
      prob.model=TRUE)

# Generate a textual view of the SVM model.

crs$ksvm

# Time taken: 0.40 secs
```

```
Support Vector Machine object of class "ksvm"

SV type: C-svc  (classification)
 parameter : cost C = 1

Gaussian Radial Basis kernel function.
 Hyperparameter : sigma =  0.264793394699108

Number of Support Vectors : 775

Objective Function Value : -122.8819 -260.3999 -64.3957 -377.883 -21.0568 -3
0.3267
Training error : 0.228041
Probability model included.
```

Multi-Class Model: Multinomial Logistic Regression


R Data Project Tools Settings Help Rattle
Mon 16:12 iss-user
R Data Miner - [Rattle (ADV_ASD_State_R.rattle)]
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Numeric Generalized Poisson Logistic Probit Multinomial
Model Builder: multinom
Plot
Linear and Generalised Linear Models
A linear regression model is the traditional method for fitting a statistical model to data. It is appropriate when the target variable is numeric and continuous.
The family of generalized linear models extends traditional linear regression to targets with non-normal (non-gaussian) distributions. Linear regression models are iteratively fit to the data after transforming the target variable to a continuous numeric.
Generalized linear regression, applied to a dataset with a numeric, continuous target variable, will build the same model, using a different algorithm.
The generalised algorithm is parameterised by the distribution of the target variable and a link function relating the mean of the target to the inputs. These two parameters describe what we often refer to as a family, such as Poisson, Logistic, etc.
If the target has just two possible outcomes it is transformed using a logistic or probit function. A probit regression gives similar results to the logistic regression, but often with smaller coefficients.
RStudio
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom Export


R Data Project Tools Settings Help Rattle
Mon 16:16 iss-user
R Data Miner - [Rattle (ADV_ASD_State_R.rattle)]
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Numeric Generalized Poisson Logistic Probit Multinomial
Model Builder: multinom
Plot
Summary of the Multinomial Regression model (built using multinom):
Call:
multinom(formula = Prevalence_Risk4 ~ ., data = crs$dataset[crs$train,
c(crs$input, crs$target)], trace = FALSE, maxit = 1000)
n=1184
Coefficients:
(Intercept)
Low -7.613185
Medium -2.609530
Very High -9.457761
Sourcemedi
Low 6.664750
Medium 2.620152
Very High 4.489966
Sourcensch
Low -10.3505526
Medium -0.7744868
Very High -2.5620704
Sourcesped
Low 7.57715
Medium 3.39157
Very High -12.32038
State_RegionD2 Middle Atlantic
Low 2.455462
Medium 1.908225
Very High 2.070621
State_RegionD3 East North Central
Low 1.659605
Medium 2.242230
Very High 1.650636
State_RegionD4 West North Central
Low 2.931481
Medium 1.732136
Very High -0.570721
State_RegionD5 South Atlantic
Low 3.22348888
Medium 2.34752178
Very High 0.02696508
State_RegionD6 East South Central
Low 5.350321
Medium 3.470543
Very High 4.747385
State_RegionD7 West South Central
Low 6.165719
Medium 4.413084
Very High -12.369540
State_RegionD8 Mountain
Low 4.41036609
Medium 2.63924552
Very High -0.06124791
State_RegionD9 Pacific
Low 3.25616127
Medium 2.30626004
The Linear model has been built. Time taken: 0.37 secs
RStudio
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom Export

In [55]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 16:13:19 x86_64-pc-linux-gnu

# Regression model

# Build a multinomial model using the nnet package.

library(nnet, quietly=TRUE)

# Summarise multinomial model using Anova from the car package.

library(car, quietly=TRUE)

# Build a Regression model.

crs$glm <- multinom(Prevalence_Risk4 ~ ., data=crs$dataset[crs$train,c(crs$inp

# Generate a textual view of the Linear model.

rattle.print.summary.multinom(summary(crs$glm,
                              Wald.ratios=TRUE))
cat(sprintf("Log likelihood: %.3f (%d df)
", logLik(crs$glm)[1], attr(logLik(crs$glm), "df")))
if (is.null(crs$glm$na.action)) omitted <- TRUE else omitted <- -crs$glm$na.ac
cat(sprintf("Pseudo R-Square: %.8f

",cor(apply(crs$glm$fitted.values, 1, function(x) which(x == max(x))),
as.integer(crs$dataset[crs$train,][omitted,]$Prevalence_Risk4))))

cat('==== ANOVA ====
')
print(Anova(crs$glm))
print("
")

# Time taken: 0.37 secs
```

```
Registered S3 methods overwritten by 'car':
  method                           from
  influence.merMod                 lme4
  cooks.distance.influence.merMod lme4
  dfbeta.influence.merMod          lme4
  dfbetas.influence.merMod         lme4

Attaching package: ‘car’

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

    logit

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

    densityPlot
```

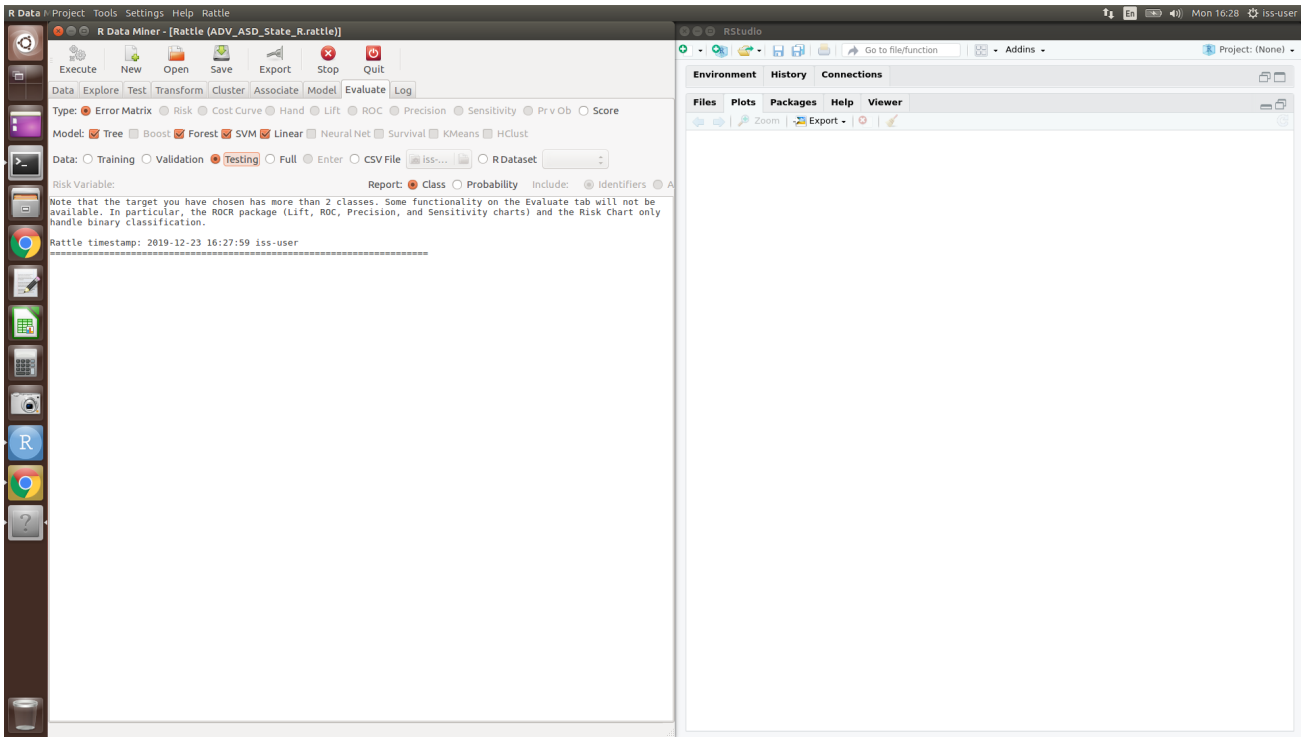
()

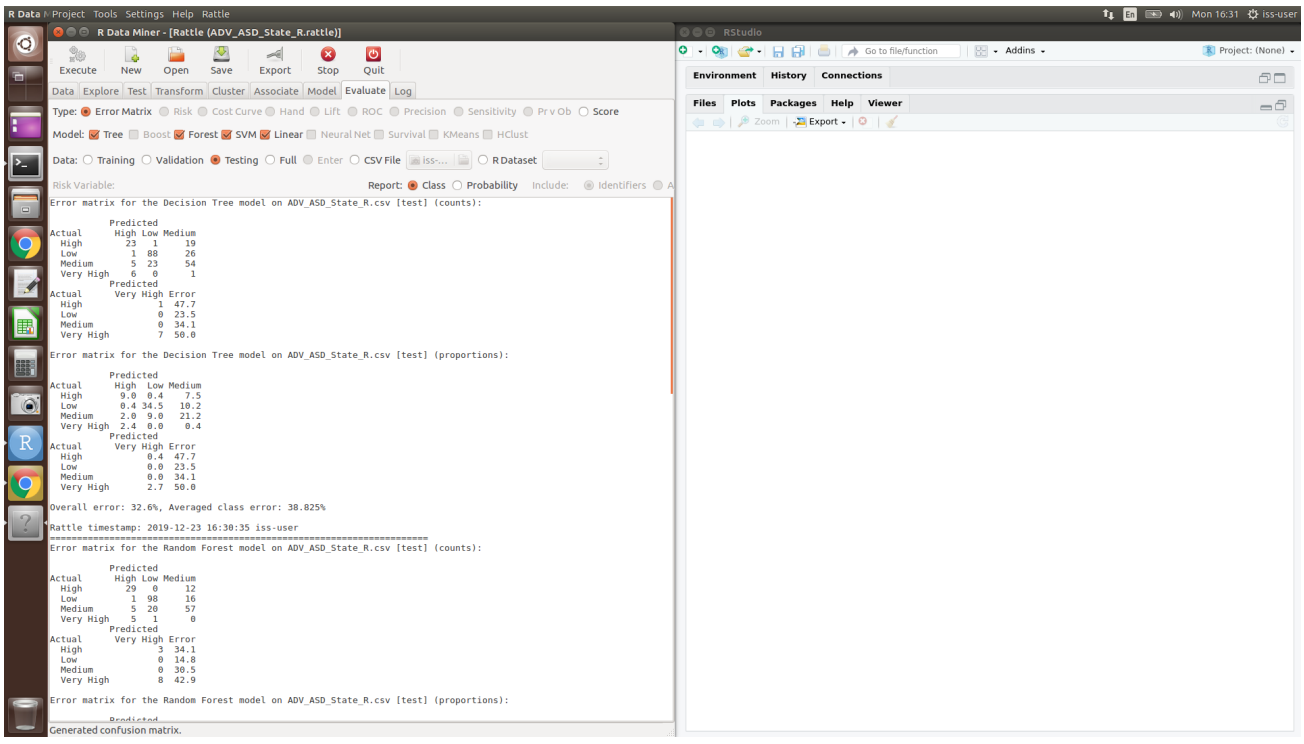
Rattle: Evaluate Model (Classification: Multi-Class)

Multi-Class Classification: Prevalence_Risk4

```
crs$rpart
crs$rf
crs$ksvm
crs$glm
```

Rattle: Evaluate Model (Classification): Error/Confusion matrix





In [56]:
```
crs$target
```

'Prevalence_Risk4'

In [57]:

```
#============================================================================
# Rattle timestamp: 2019-12-23 16:32:12 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# Generate an Error Matrix for the Decision Tree model.

# Obtain the response from the Decision Tree model.

crs$pr <- predict(crs$rpart, newdata=crs$dataset[crs$test, c(crs$input, crs$ta
    type="class")

# Generate the confusion matrix showing counts.

rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalence

# Generate the confusion matrix showing proportions.

(per <- rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Pr

# Calculate the overall error percentage.

cat(100-sum(diag(per), na.rm=TRUE))

# Calculate the averaged class error percentage.

cat(mean(per[,"Error"], na.rm=TRUE))

# Generate an Error Matrix for the Random Forest model.

# Obtain the response from the Random Forest model.

crs$pr <- predict(crs$rf, newdata=na.omit(crs$dataset[crs$test, c(crs$input, c

# Generate the confusion matrix showing counts.

rattle::errorMatrix(na.omit(crs$dataset[crs$test, c(crs$input, crs$target)])$P

# Generate the confusion matrix showing proportions.

(per <- rattle::errorMatrix(na.omit(crs$dataset[crs$test, c(crs$input, crs$tar

# Calculate the overall error percentage.

cat(100-sum(diag(per), na.rm=TRUE))

# Calculate the averaged class error percentage.

cat(mean(per[,"Error"], na.rm=TRUE))

# Generate an Error Matrix for the SVM model.

# Obtain the response from the SVM model.

crs$pr <- kernlab::predict(crs$ksvm, newdata=na.omit(crs$dataset[crs$test, c(c

# Generate the confusion matrix showing counts.

rattle::errorMatrix(na.omit(crs$dataset[crs$test, c(crs$input, crs$target)])$P

# Generate the confusion matrix showing proportions.

(per <- rattle::errorMatrix(na.omit(crs$dataset[crs$test, c(crs$input, crs$tar
```

```
# Calculate the overall error percentage.

cat(100-sum(diag(per), na.rm=TRUE))

# Calculate the averaged class error percentage.

cat(mean(per[,"Error"], na.rm=TRUE))

# Generate an Error Matrix for the Linear model.

# Obtain the response from the Linear model.

crs$pr <- predict(crs$glm, newdata=crs$dataset[crs$test, c(crs$input, crs$targ

# Generate the confusion matrix showing counts.

rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalence

# Generate the confusion matrix showing proportions.

(per <- rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Pr

# Calculate the overall error percentage.

cat(100-sum(diag(per), na.rm=TRUE))

# Calculate the averaged class error percentage.

cat(mean(per[,"Error"], na.rm=TRUE))
```

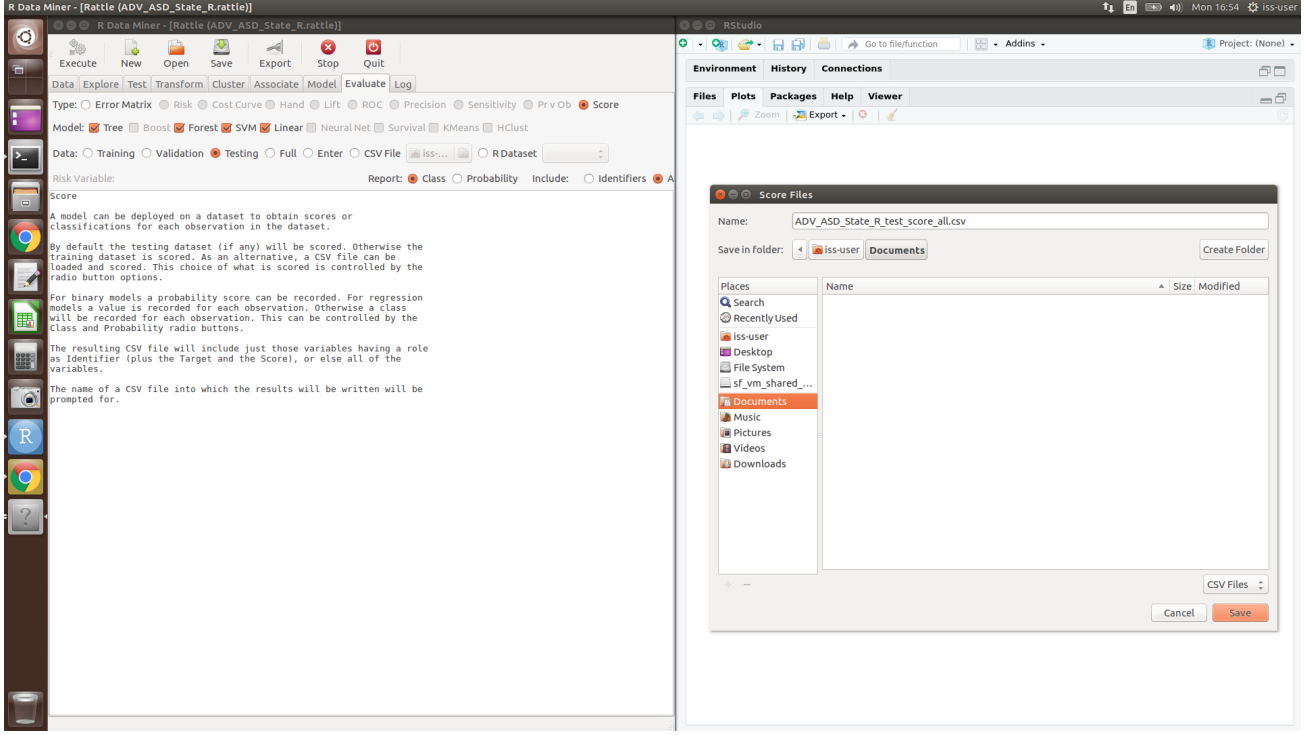
	High	Low	Medium	Very High	Error
High	23	1	19	1	47.7
Low	1	88	26	0	23.5
Medium	5	23	54	0	34.1
Very High	6	0	1	7	50.0

	High	Low	Medium	Very High	Error
High	9.0	0.4	7.5	0.4	47.7
Low	0.4	34.5	10.2	0.0	23.5
Medium	2.0	9.0	21.2	0.0	34.1
Very High	2.4	0.0	0.4	2.7	50.0

32.638.825

High	Low	Medium	Very High	Error

Rattle: Evaluate Model (Classification): Score/Write predicted results to CSV file.

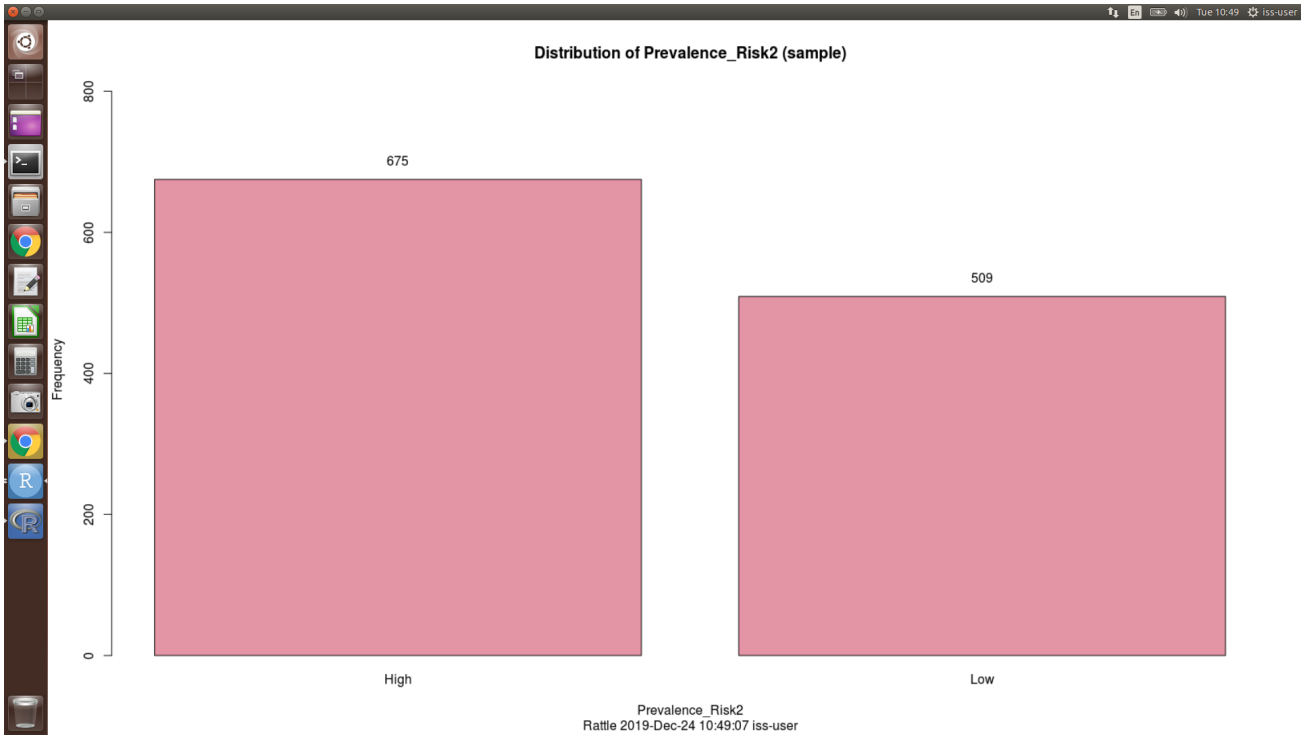


In [58]:

```
#====================================================================
# Rattle timestamp: 2019-12-23 16:50:20 x86_64-pc-linux-gnu

# Score the testing dataset.

# Obtain probability scores for the Decision Tree model on ADV_ASD_State_R.csv

crs$pr_rpart <- predict(crs$rpart, newdata=crs$dataset[crs$test, c(crs$input)]
    type="class")

# Obtain probability scores for the Random Forest model on ADV_ASD_State_R.csv

crs$pr_rf <- predict(crs$rf, newdata=na.omit(crs$dataset[crs$test, c(crs$input

# Obtain probability scores for the SVM model on ADV_ASD_State_R.csv [test].

crs$pr_ksvm <- kernlab::predict(crs$ksvm, newdata=na.omit(crs$dataset[crs$test

# Obtain probability scores for the Linear model on ADV_ASD_State_R.csv [test]

crs$pr_glm <- predict(crs$glm, newdata=crs$dataset[crs$test, c(crs$input)])

# Extract the relevant variables from the dataset.

sdata <- crs$dataset[crs$test,]

# Output the combined data.

write.csv(cbind(sdata, crs$pr_rpart, crs$pr_rf, crs$pr_ksvm, crs$pr_glm), row.
          file="../reference/R rattle/ADV_ASD_State_R_test_score_all_Prevalenc
```

Rattle: Train Model (Classification)

Binary-Class Classification: Prevalence_Risk2


Tue 10:49
iss-user
Distribution of Prevalence_Risk2 (sample)
Frequency
0
200
400
600
800
675
509
High
Low
Prevalence_Risk2
Rattle 2019-Dec-24 10:49:07 iss-user

In [59]:

```
#=====================================================================
# Rattle timestamp: 2019-12-24 10:52:36 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# Build the train/validate/test datasets.

# nobs=1692 train=1184 validate=254 test=254

set.seed(88)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test

# The following variable selections have been noted.

crs$input     <- c("Source", "State_Region", "RMD_Year",
                   "RMD_R10_Denominator")

crs$numeric   <- c("RMD_Year", "RMD_R10_Denominator")

crs$categoric <- c("Source", "State_Region")

crs$target    <- "Prevalence_Risk2"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- c("State", "Denominator", "Prevalence", "Lower.CI", "Upper.CI
crs$weights   <- NULL
```

In [60]:

```
crs$target
```

'Prevalence_Risk2'

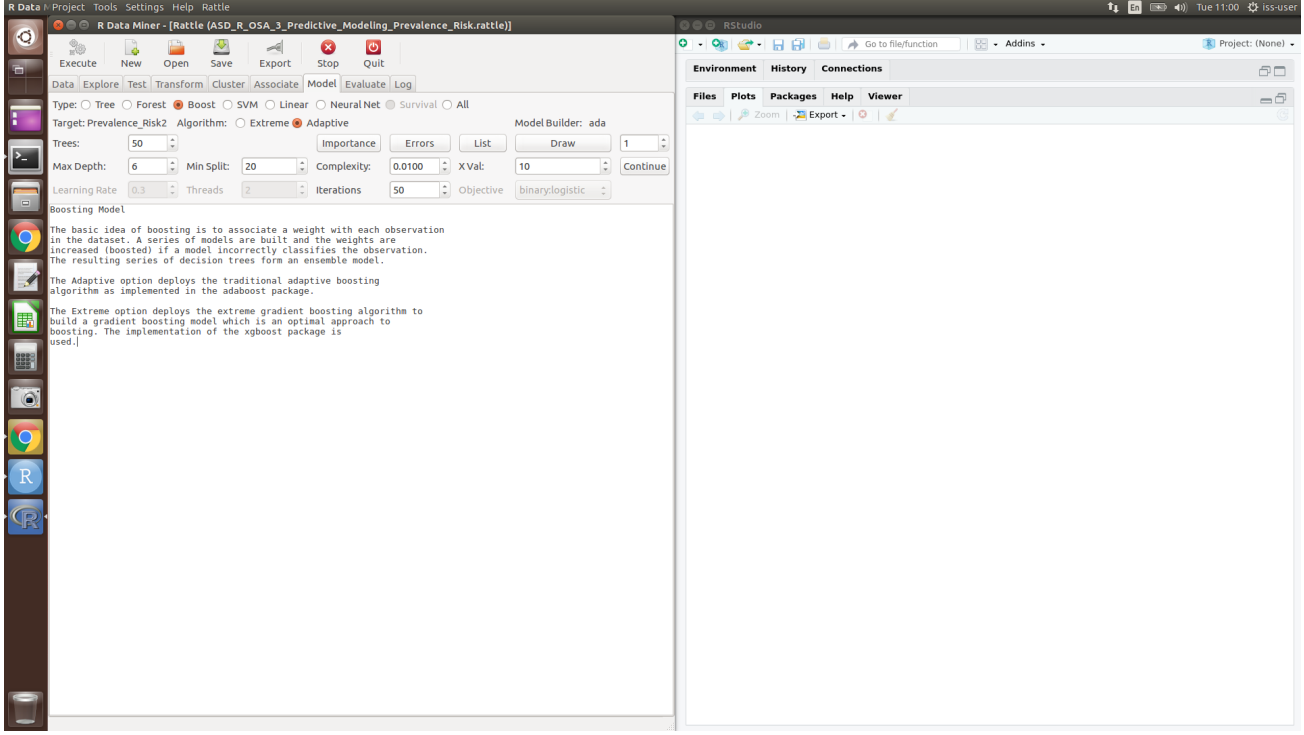
In [61]:

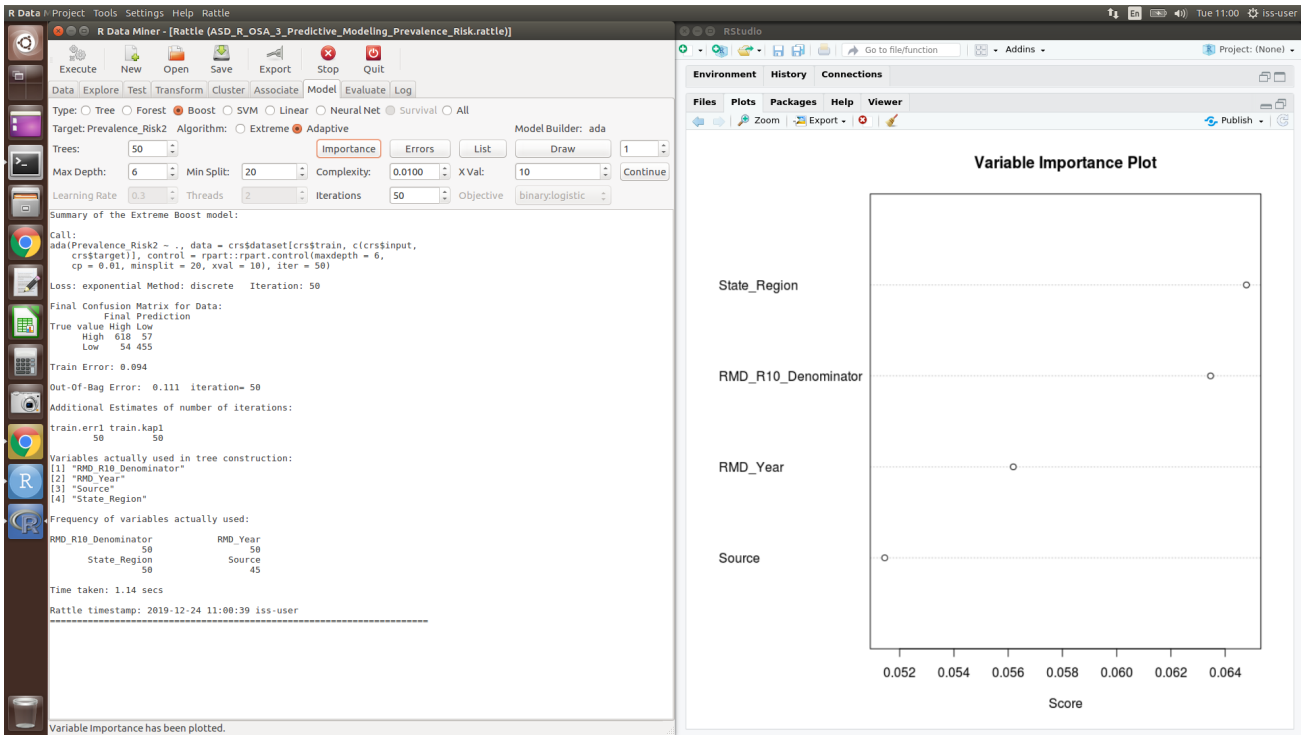
```
crs$input
```

'Source' 'State_Region' 'RMD_Year' 'RMD_R10_Denominator'

Binary-Class Model: Boost

Ada Boost: Adaptive Boosting





In [62]:
```
if(!require(ada)){install.packages("ada")}
library('ada')
```

```
Loading required package: ada
```

In [63]:

```
#============================================================================
# Rattle timestamp: 2019-12-24 11:00:37 x86_64-pc-linux-gnu

# Ada Boost

# The `ada' package implements the boost algorithm.

# Build the Ada Boost model.

set.seed(crv$seed)
crs$ada <- ada::ada(Prevalence_Risk2 ~ .,
                    data=crs$dataset[crs$train,c(crs$input, crs$target)],
                    control=rpart::rpart.control(maxdepth=6,
                                                 cp=0.010000,
                                                 minsplit=20,
                                                 xval=10),
                    iter=50)

# Print the results of the modelling.

print(crs$ada)
round(crs$ada$model$errs[crs$ada$iter,], 2)
cat('Variables actually used in tree construction:\n')
print(sort(names(listAdaVarsUsed(crs$ada))))
cat('\nFrequency of variables actually used:\n')
print(listAdaVarsUsed(crs$ada))

# Time taken: 1.14 secs

# Plot the error rate as we increase the number of trees.

plot(crs$ada)

# Plot the relative importance of the variables.

ada::varplot(crs$ada)
```

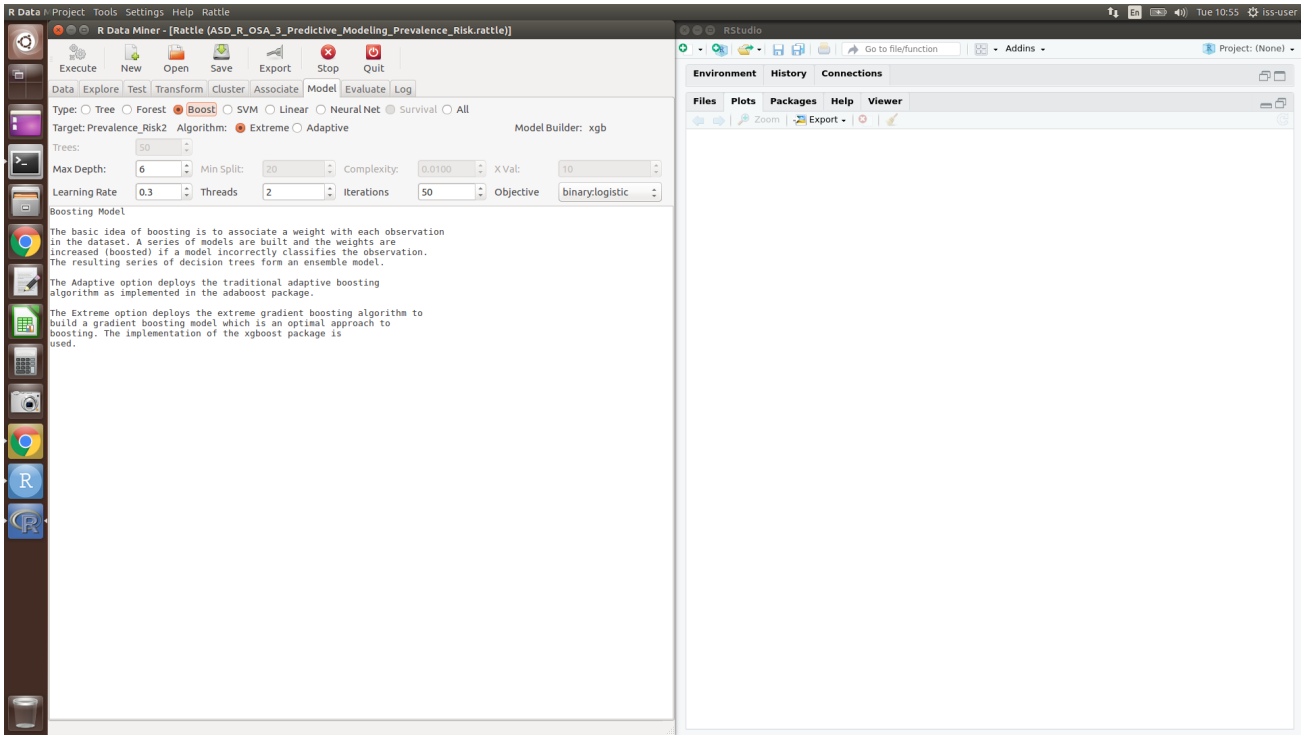
```
Call:
ada(Prevalence_Risk2 ~ ., data = crs$dataset[crs$train, c(crs$input,
    crs$target)], control = rpart::rpart.control(maxdepth = 6,
    cp = 0.01, minsplit = 20, xval = 10), iter = 50)

Loss: exponential Method: discrete   Iteration: 50

Final Confusion Matrix for Data:
          Final Prediction
True value High Low
      High  618  57
      Low    63 446

Train Error: 0.101

Out-Of-Bag Error:  0.112  iteration= 45

Additional Estimates of number of iterations:

train.err1 train.kap1
```

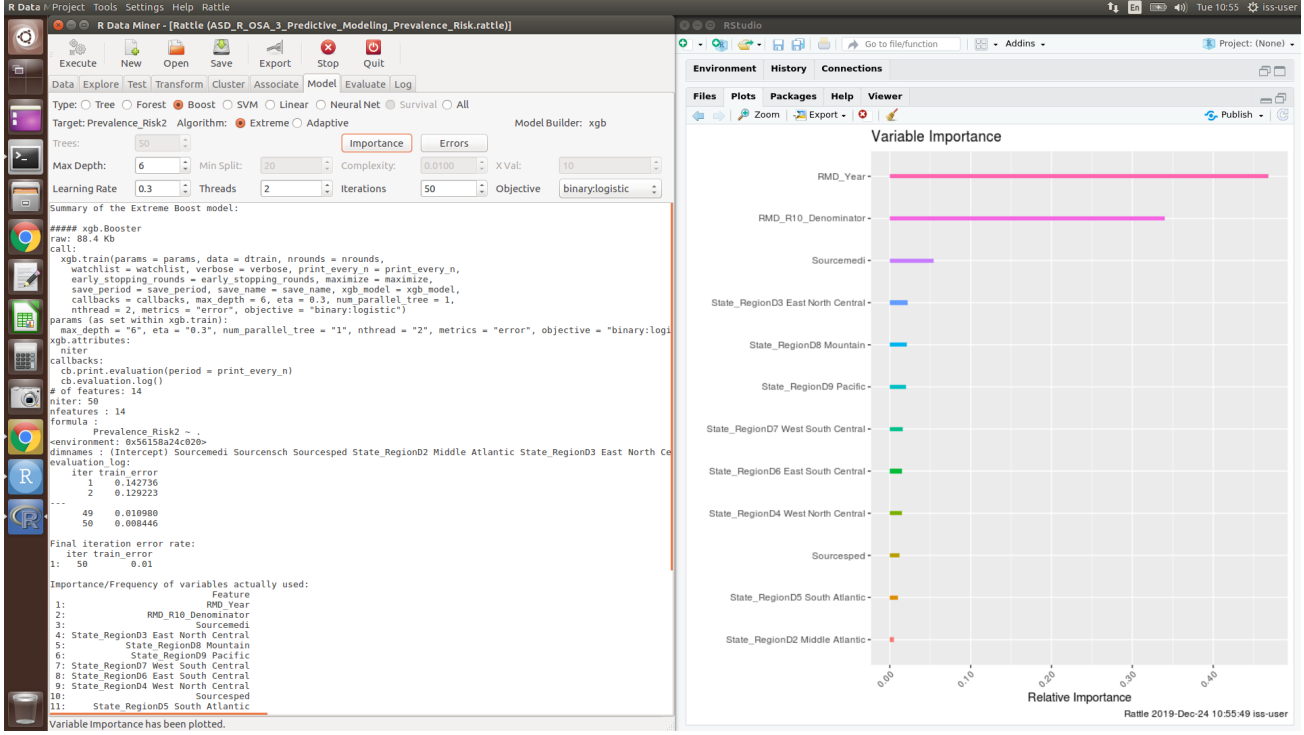
In [64]:

```
# Display tree number 1.

listTreesAda(crs$ada, 1)
```

```
Tree 1 of 50:
n= 592

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 592 0.2162162000 -1 (0.63271162 0.36728838)
   2) RMD_Year>=-0.252934 384 0.0692567600 -1 (0.82858636 0.17141364)
     4) RMD_Year>=0.4215567 230 0.0143581100 -1 (0.94267650 0.05732350) *
     5) RMD_Year< 0.4215567 154 0.0548986500 -1 (0.64248883 0.35751117)
      10) State_Region=D1 New England,D2 Middle Atlantic,D3 East North Centra
l,D5 South Atlantic 71 0.0109797300 -1 (0.85413745 0.14586255) *
      11) State_Region=D4 West North Central,D6 East South Central,D7 West So
uth Central,D8 Mountain,D9 Pacific 83 0.0261824300 1 (0.43897505 0.56102495)

        22) State_Region=D4 West North Central,D9 Pacific 38 0.0152027000 -1
(0.59322034 0.40677966)
          44) RMD_R10_Denominator>=-0.738634 24 0.0067567570 -1 (0.72413793
0.27586207) *
          45) RMD_R10_Denominator< -0.738634 14 0.0033783780 1 (0.34426230 0.
65573770) *
        23) State_Region=D6 East South Central,D7 West South Central,D8 Mount
ain 45 0.0092905410 1 (0.29806452 0.70193548) *
   3) RMD_Year< -0.252934 208 0.0287162200 1 (0.20411664 0.79588336)
     6) Source=addm,nsch 9 0.0008445946 -1 (0.91304348 0.08695652) *
     7) Source=medi,sped 199 0.0219594600 1 (0.16475558 0.83524442) *
```

Xg Boost: Extreme Gradient Boost





In [65]:
```
if(!require(xgboost)){install.packages("xgboost")}
library('xgboost')
```

```
Loading required package: xgboost

Attaching package: ‘xgboost’

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

    xgboost
```

```
#============================================================================
# Rattle timestamp: 2019-12-24 11:21:50 x86_64-pc-linux-gnu

# Extreme Boost

# The `xgboost' package implements the extreme gradient boost algorithm.

# Build the Extreme Boost model.

set.seed(crv$seed)

crs$xgb <- xgboost(Prevalence_Risk2 ~ .,
  data              = crs$dataset[crs$train,c(crs$input, crs$target)],
  max_depth         = 6,
  eta               = 0.3,
  num_parallel_tree = 1,
  nthread           = 2,
  nround            = 50,
  metrics           = 'error',
  objective         = 'binary:logistic')

# Print the results of the modelling.

print(crs$xgb)

cat('\nFinal iteration error rate:\n')
print(round(crs$xgb$evaluation_log[crs$xgb$niter, ], 2))

cat('\nImportance/Frequency of variables actually used:\n')
```

```
print(crs$imp <- importance(crs$xgb, crs$dataset[crs$train,c(crs$input,
crs$target)]))

# Time taken: 0.10 secs
```

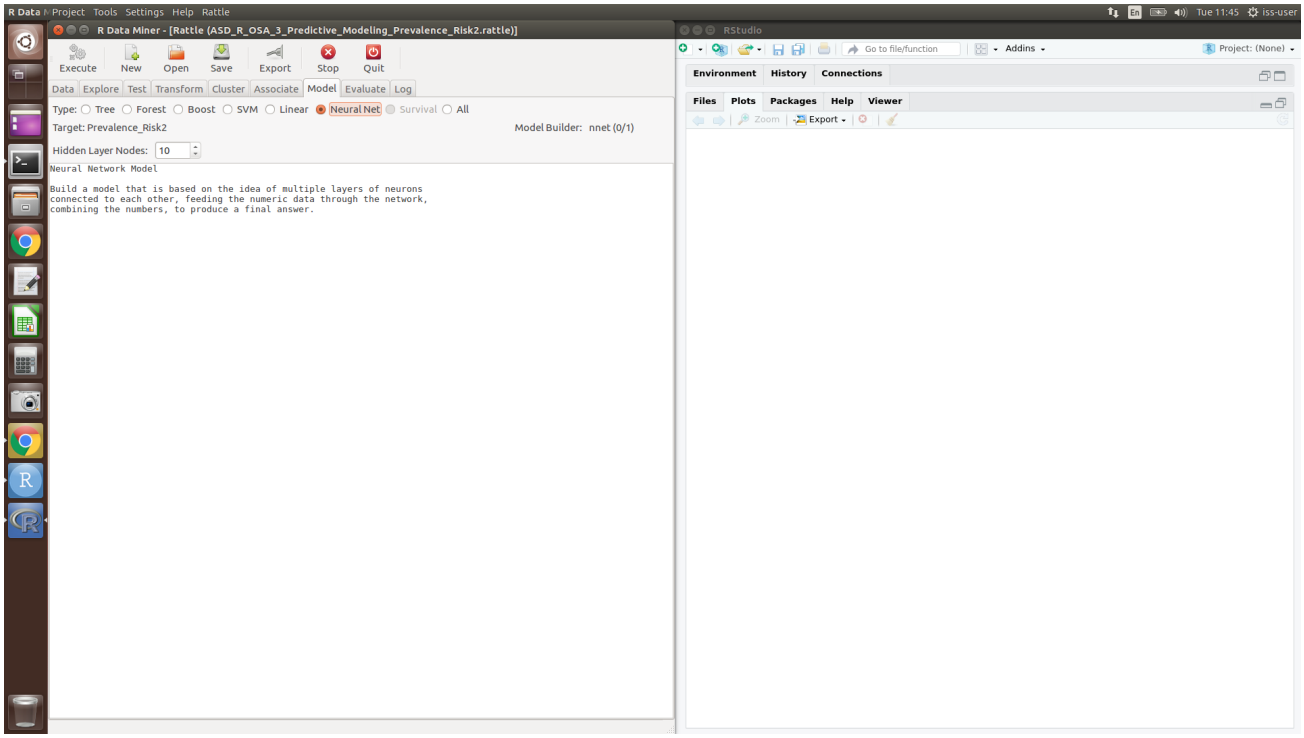
Note: Above R code generated from Rattle log cannot be executed directly due to: "Error in xgb.get.DMatrix(data, label, missing, weight): xgboost doesn't support data.frame as input. Convert it to matrix first."

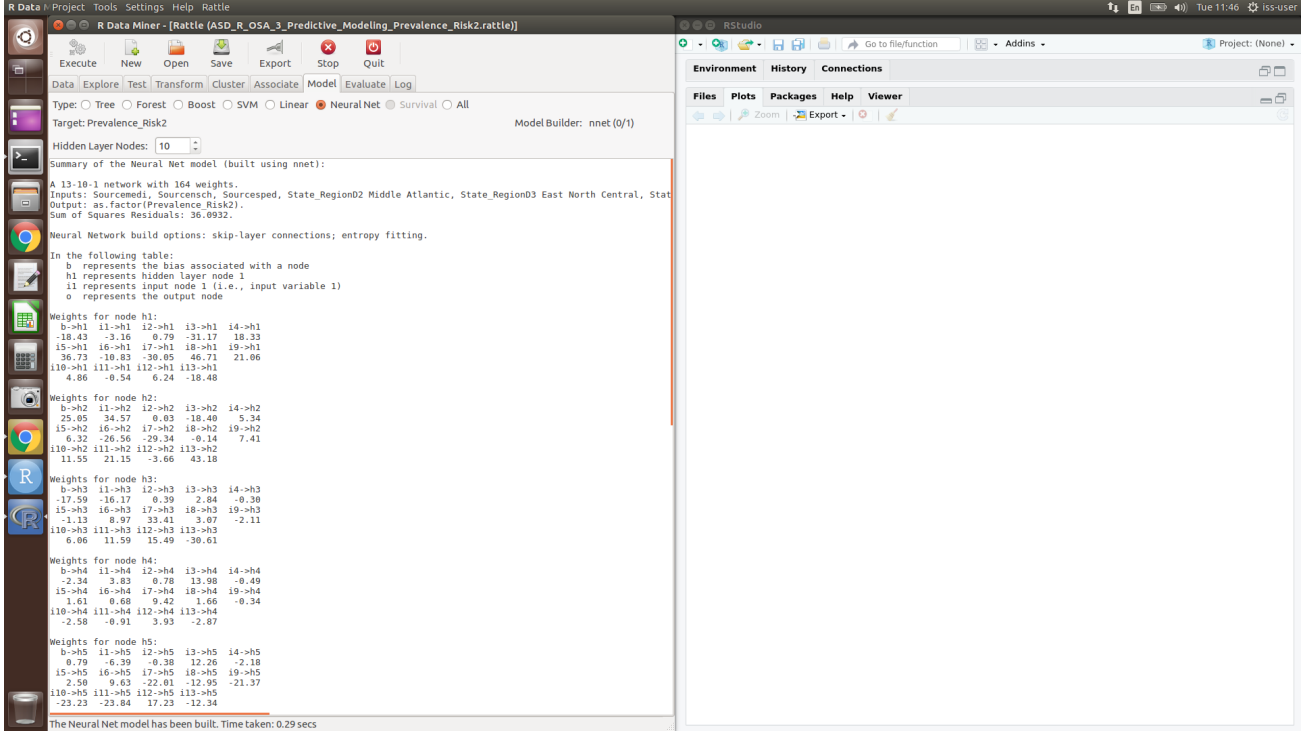
In the workshop submission section, you are required to build **xgboost** model by fixing/handling this issue. References:

https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/HousePriceAnalysisPrediction/codeR/House%20prices_%20Lasso%2C%20XGBoost%2C%
(https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/HousePriceAnalysisPrediction/codeR/House%20prices_%20Lasso%2C%20XGBoost%2C%

https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/Google%20Analytics%20Customer%20Revenue%20Prediction/code/Google%20Analytics%
(https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/Google%20Analytics%20Customer%20Revenue%20Prediction/code/Google%20Analytics%

Binary-Class Model: Neural Net (NN)





In [66]:
```
if(!require(nnet)){install.packages("nnet")}
library('nnet')
```

In [67]:

```
#=====================================================================
# Rattle timestamp: 2019-12-24 11:46:00 x86_64-pc-linux-gnu

# Neural Network

# Build a neural network model using the nnet package.

library(nnet, quietly=TRUE)

# Build the NNet model.

set.seed(199)
crs$nnet <- nnet(as.factor(Prevalence_Risk2) ~ .,
    data=crs$dataset[crs$train,c(crs$input, crs$target)],
    size=10, skip=TRUE, MaxNWts=10000, trace=FALSE, maxit=100)

# Print the results of the modelling.

cat(sprintf("A %s network with %d weights.\n",
    paste(crs$nnet$n, collapse="-"),
    length(crs$nnet$wts)))
cat(sprintf("Inputs: %s.\n",
    paste(crs$nnet$coefnames, collapse=", ")))
cat(sprintf("Output: %s.\n",
    names(attr(crs$nnet$terms, "dataClasses"))[1]))
cat(sprintf("Sum of Squares Residuals: %.4f.\n",
    sum(residuals(crs$nnet) ^ 2)))
cat("\n")
print(summary(crs$nnet))
cat('\n')

# Time taken: 0.29 secs
```

```
A 13-10-1 network with 164 weights.
Inputs: Sourcemedi, Sourcensch, Sourcesped, State_RegionD2 Middle Atlantic,
State_RegionD3 East North Central, State_RegionD4 West North Central, State
_RegionD5 South Atlantic, State_RegionD6 East South Central, State_RegionD7
West South Central, State_RegionD8 Mountain, State_RegionD9 Pacific, RMD_Ye
ar, RMD_R10_Denominator.
Output: as.factor(Prevalence_Risk2).
Sum of Squares Residuals: 36.0932.

a 13-10-1 network with 164 weights
options were - skip-layer connections  entropy fitting
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9
->h1
 -18.43   -3.16    0.79  -31.17   18.33   36.73  -10.83  -30.05   46.71   2
1.06
i10->h1 i11->h1 i12->h1 i13->h1
   4.86   -0.54    6.24  -18.48
  b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9
->h2
```

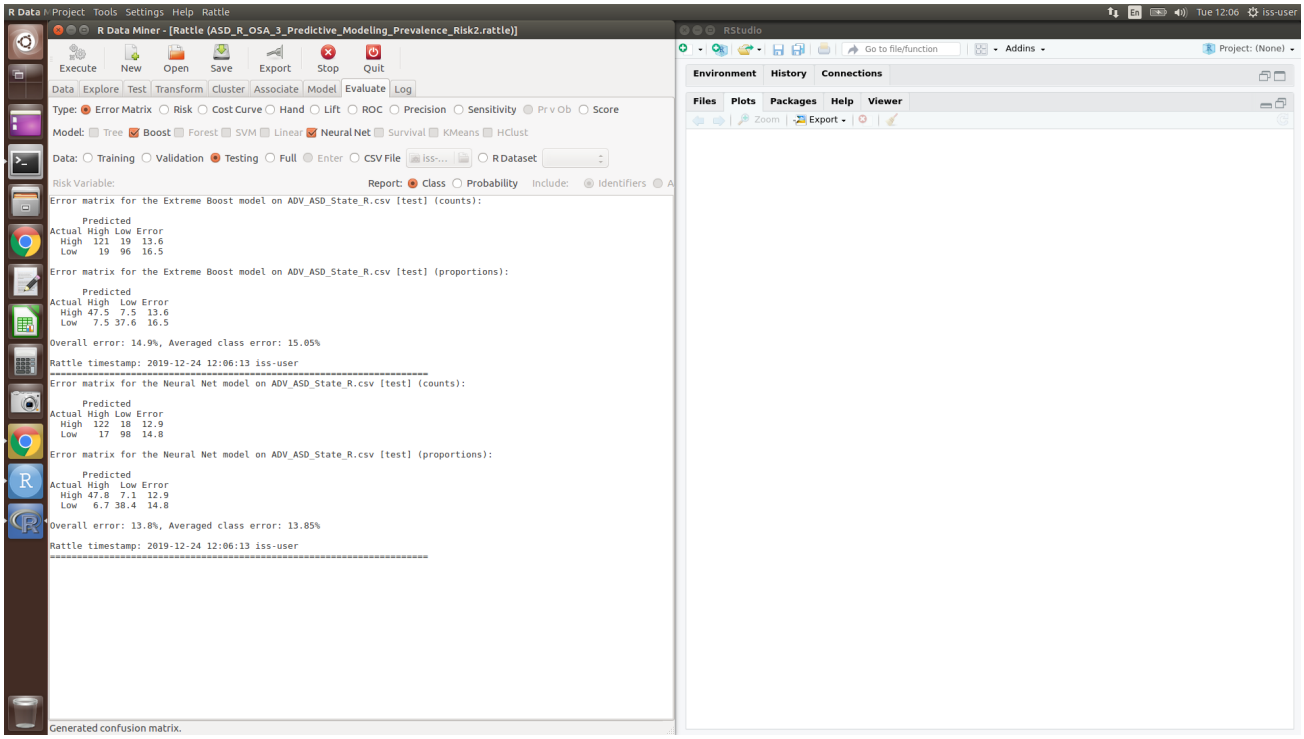
()

Rattle: Evaluate Model (Classification: Binary-Class)

Binary-Class Classification: Prevalence_Risk2

```
crs$ada
crs$nnet
```

Rattle: Evaluate Model (Classification): Error/Confusion matrix



In [68]:

```
#=====================================================================
# Rattle timestamp: 2019-12-24 12:07:18 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# Generate an Error Matrix for the Extreme Boost model.

# Obtain the response from the Extreme Boost model.

crs$pr <- predict(crs$ada, newdata=crs$dataset[crs$test, c(crs$input, crs$targ

# Generate the confusion matrix showing counts.

rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalence

# Generate the confusion matrix showing proportions.

(per <- rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Pr

# Calculate the overall error percentage.

cat(100-sum(diag(per), na.rm=TRUE))

# Calculate the averaged class error percentage.

cat(mean(per[,"Error"], na.rm=TRUE))

# Generate an Error Matrix for the Neural Net model.

# Obtain the response from the Neural Net model.

crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input, crs$tar

# Generate the confusion matrix showing counts.

rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalence

# Generate the confusion matrix showing proportions.

(per <- rattle::errorMatrix(crs$dataset[crs$test, c(crs$input, crs$target)]$Pr

# Calculate the overall error percentage.

cat(100-sum(diag(per), na.rm=TRUE))

# Calculate the averaged class error percentage.

cat(mean(per[,"Error"], na.rm=TRUE))
```

	High	Low	Error
High	119	21	15.0
Low	22	93	19.1

	High	Low	Error
High	46.7	8.2	15.0
Low	8.6	36.5	19.1

16.817.05

	High	Low	Error

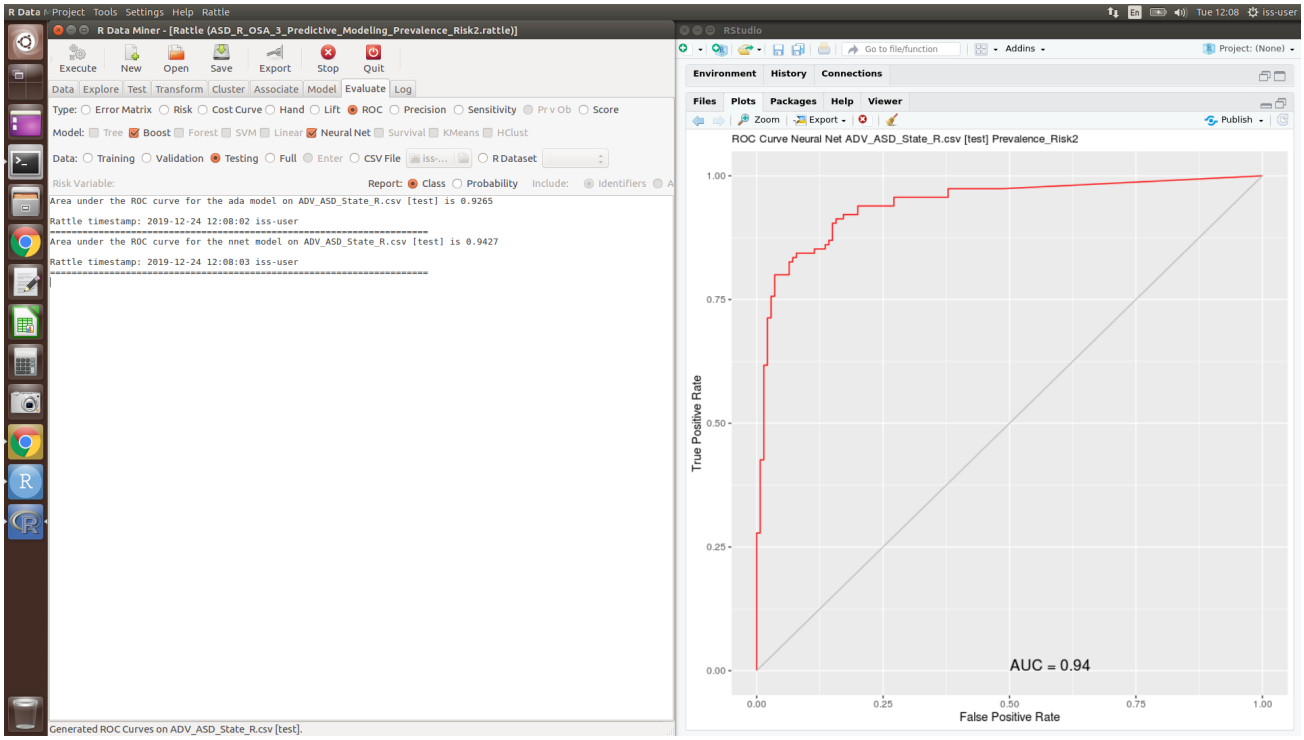
	High	Low	Error
High	122	18	12.9
Low	17	98	14.8

	High	Low	Error
High	47.8	7.1	12.9
Low	6.7	38.4	14.8

13.813.85

Rattle: Evaluate Model (Classification): ROC

ROC Curve: **TPR (True Positive Rate) / Hit Rate / Recall / Sensitivity** vs. **FPR / 1- Specificity**



In [69]:
```
if(!require(ROCR)){install.packages("ROCR")}
library('ROCR')
```

```
Loading required package: ROCR
```

In [70]:

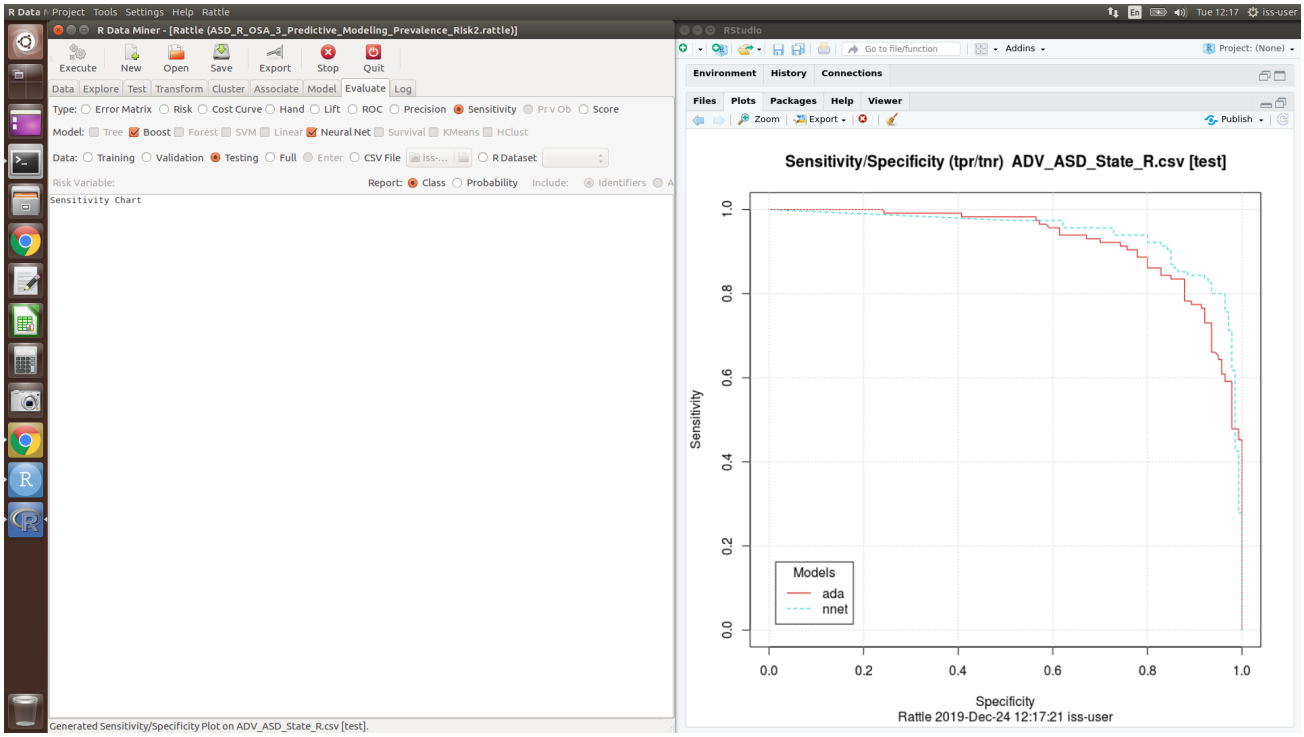
```
#=====================================================================
# Rattle timestamp: 2019-12-24 12:09:03 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# ROC Curve: requires the ROCR package.

library(ROCR)

# ROC Curve: requires the ggplot2 package.

library(ggplot2, quietly=TRUE)

# Generate an ROC Curve for the ada model on ADV_ASD_State_R.csv [test].

crs$pr <- predict(crs$ada, newdata=crs$dataset[crs$test, c(crs$input, crs$targ

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}

pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Extreme Boost ADV_ASD_State_R.csv [test] Prevalenc
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)

# Calculate the area under the curve for the plot.


# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")

# ROC Curve: requires the ROCR package.
```

```
library(ROCR)

# ROC Curve: requires the ggplot2 package.

library(ggplot2, quietly=TRUE)

# Generate an ROC Curve for the nnet model on ADV_ASD_State_R.csv [test].

crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input, crs$tar

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}

pe <- performance(pred, "tpr", "fpr")
au <- performance(pred, "auc")@y.values[[1]]
pd <- data.frame(fpr=unlist(pe@x.values), tpr=unlist(pe@y.values))
p <- ggplot(pd, aes(x=fpr, y=tpr))
p <- p + geom_line(colour="red")
p <- p + xlab("False Positive Rate") + ylab("True Positive Rate")
p <- p + ggtitle("ROC Curve Neural Net ADV_ASD_State_R.csv [test] Prevalence_R
p <- p + theme(plot.title=element_text(size=10))
p <- p + geom_line(data=data.frame(), aes(x=c(0,1), y=c(0,1)), colour="grey")
p <- p + annotate("text", x=0.50, y=0.00, hjust=0, vjust=0, size=5,
                  label=paste("AUC =", round(au, 2)))
print(p)

# Calculate the area under the curve for the plot.


# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
performance(pred, "auc")
```

```
An object of class "performance"
Slot "x.name":
[1] "None"

Slot "y.name":
[1] "Area under the ROC curve"

Slot "alpha.name":
[1] "none"
```

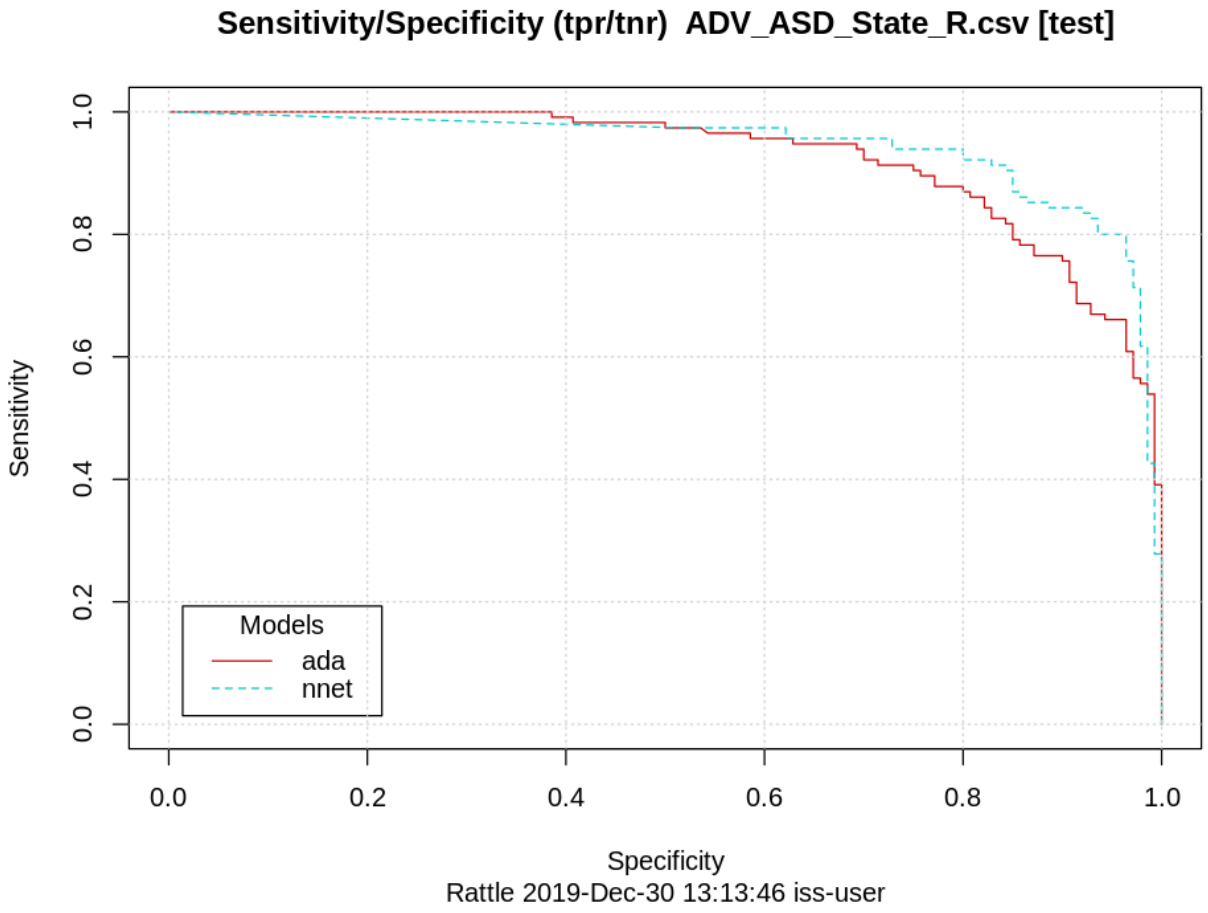
```
Slot "x.values":
list()

Slot "y.values":
[[1]]
[1] 0.9240062


Slot "alpha.values":
```

Rattle: Evaluate Model (Classification): Sensitivity

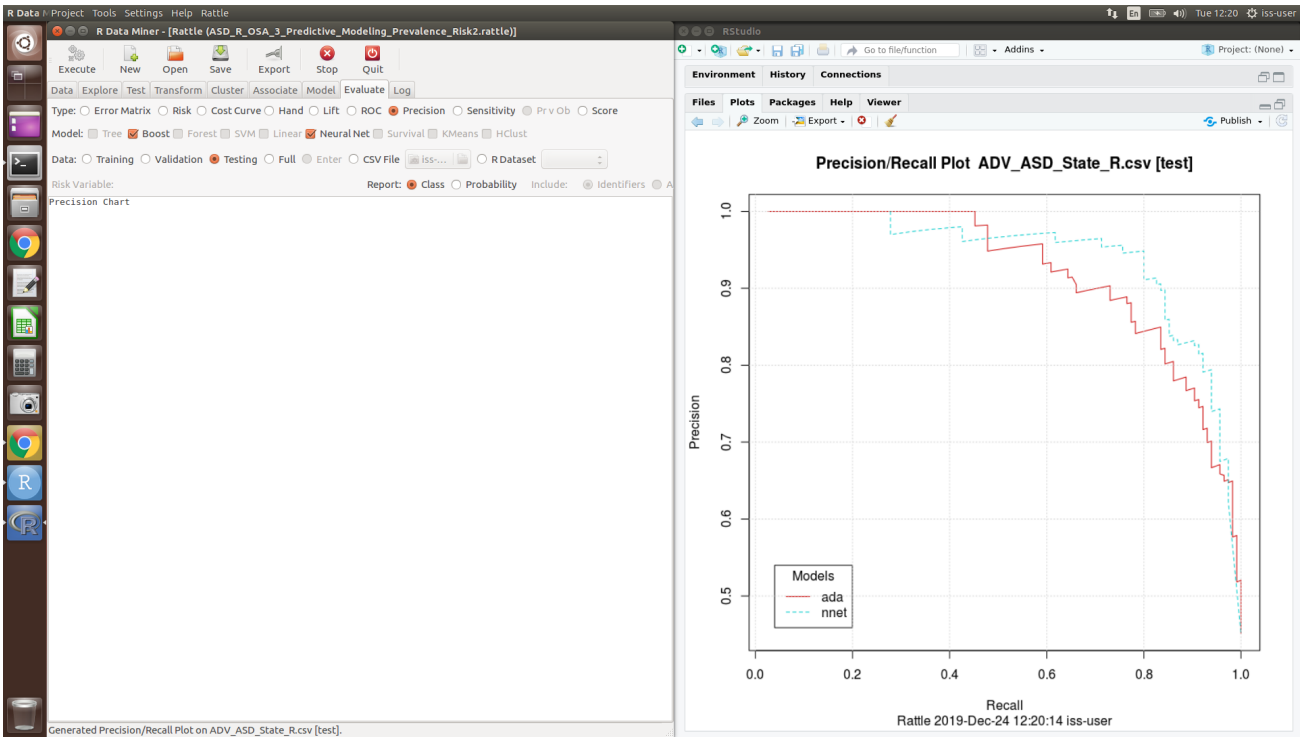


In [71]:

```
#=============================================================================
# Rattle timestamp: 2019-12-24 12:17:21 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# Sensitivity/Specificity Plot: requires the ROCR package

library(ROCR)

# Generate Sensitivity/Specificity Plot for ada model on ADV_ASD_State_R.csv [

crs$pr <- predict(crs$ada, newdata=crs$dataset[crs$test, c(crs$input, crs$targ

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
ROCR::plot(performance(pred, "sens", "spec"), col="#CC0000FF", lty=1, add=FALS


# Sensitivity/Specificity Plot: requires the ROCR package

library(ROCR)

# Generate Sensitivity/Specificity Plot for nnet model on ADV_ASD_State_R.csv

crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input, crs$tar

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
ROCR::plot(performance(pred, "sens", "spec"), col="#00CCCCFF", lty=2, add=TRUE


# Add a legend to the plot.

legend("bottomleft", c("ada","nnet"), col=rainbow(2, 1, .8), lty=1:2, title="M

# Add decorations to the plot.

title(main="Sensitivity/Specificity (tpr/tnr)  ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()
```

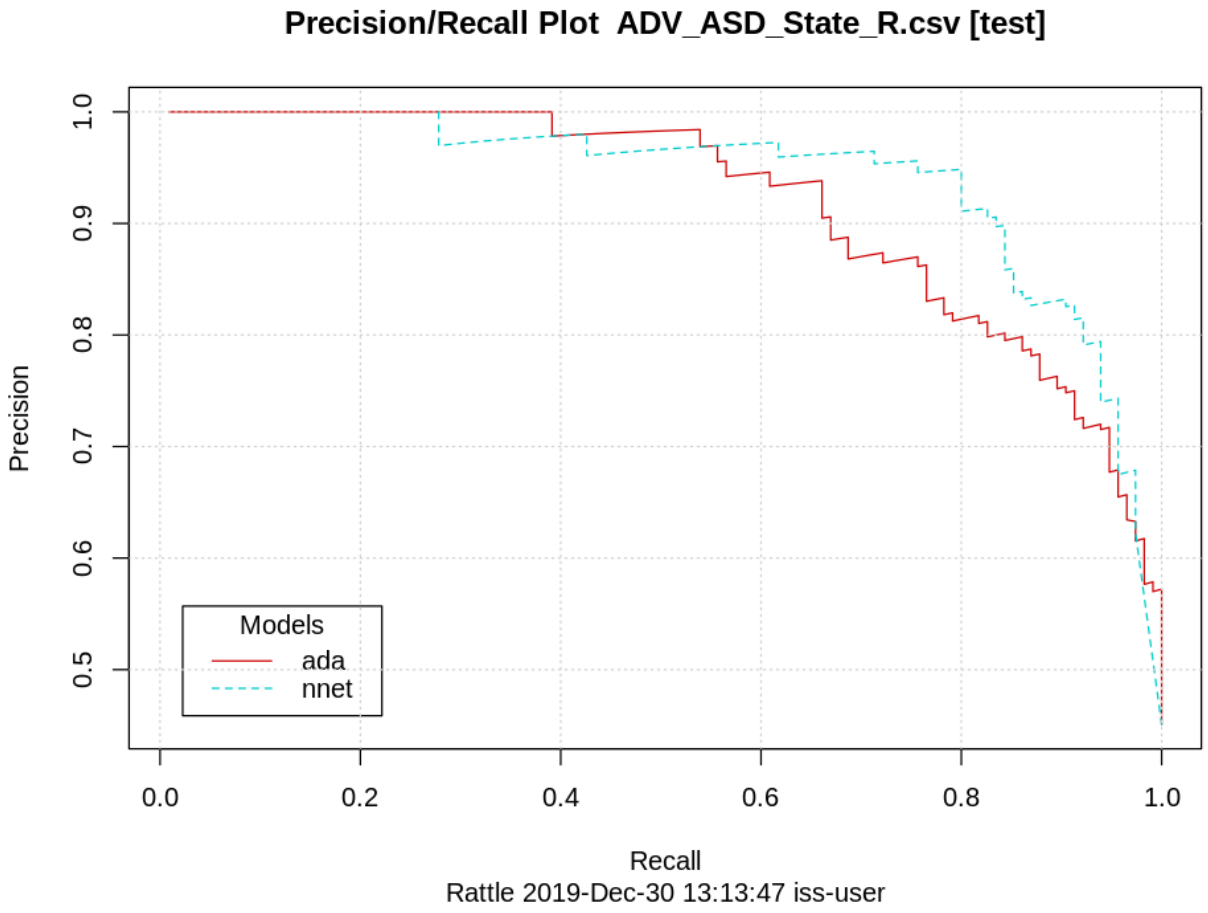


Rattle: Evaluate Model (Classification): Precision

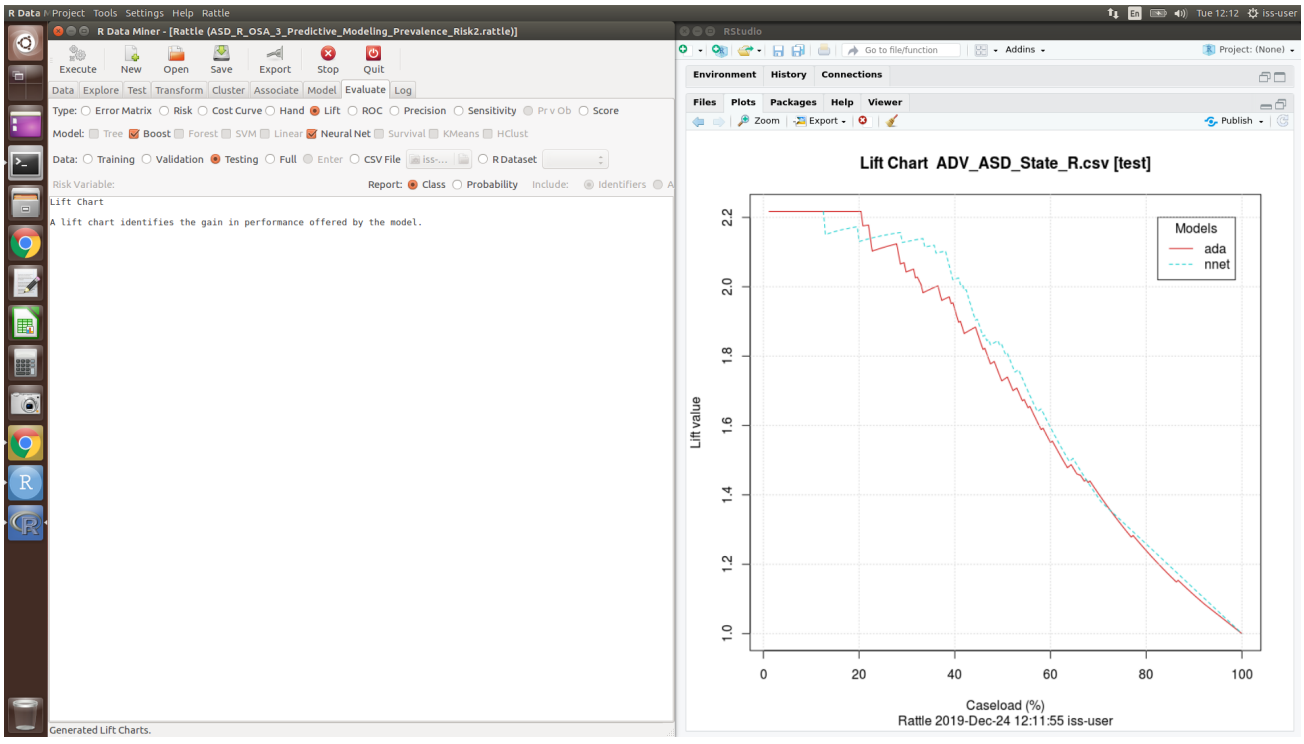


In [72]:

```
#=======================================================================
# Rattle timestamp: 2019-12-24 12:20:14 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# Precision/Recall Plot: requires the ROCR package

library(ROCR)

# Generate a Precision/Recall Plot for the ada model on ADV_ASD_State_R.csv [t

crs$pr <- predict(crs$ada, newdata=crs$dataset[crs$test, c(crs$input, crs$targ

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
ROCR::plot(performance(pred, "prec", "rec"), col="#CC0000FF", lty=1, add=FALSE


# Precision/Recall Plot: requires the ROCR package

library(ROCR)

# Generate a Precision/Recall Plot for the nnet model on ADV_ASD_State_R.csv [

crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input, crs$tar

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}
ROCR::plot(performance(pred, "prec", "rec"), col="#00CCCCFF", lty=2, add=TRUE)


# Add a legend to the plot.

legend("bottomleft", c("ada","nnet"), col=rainbow(2, 1, .8), lty=1:2, title="M

# Add decorations to the plot.

title(main="Precision/Recall Plot  ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()
```



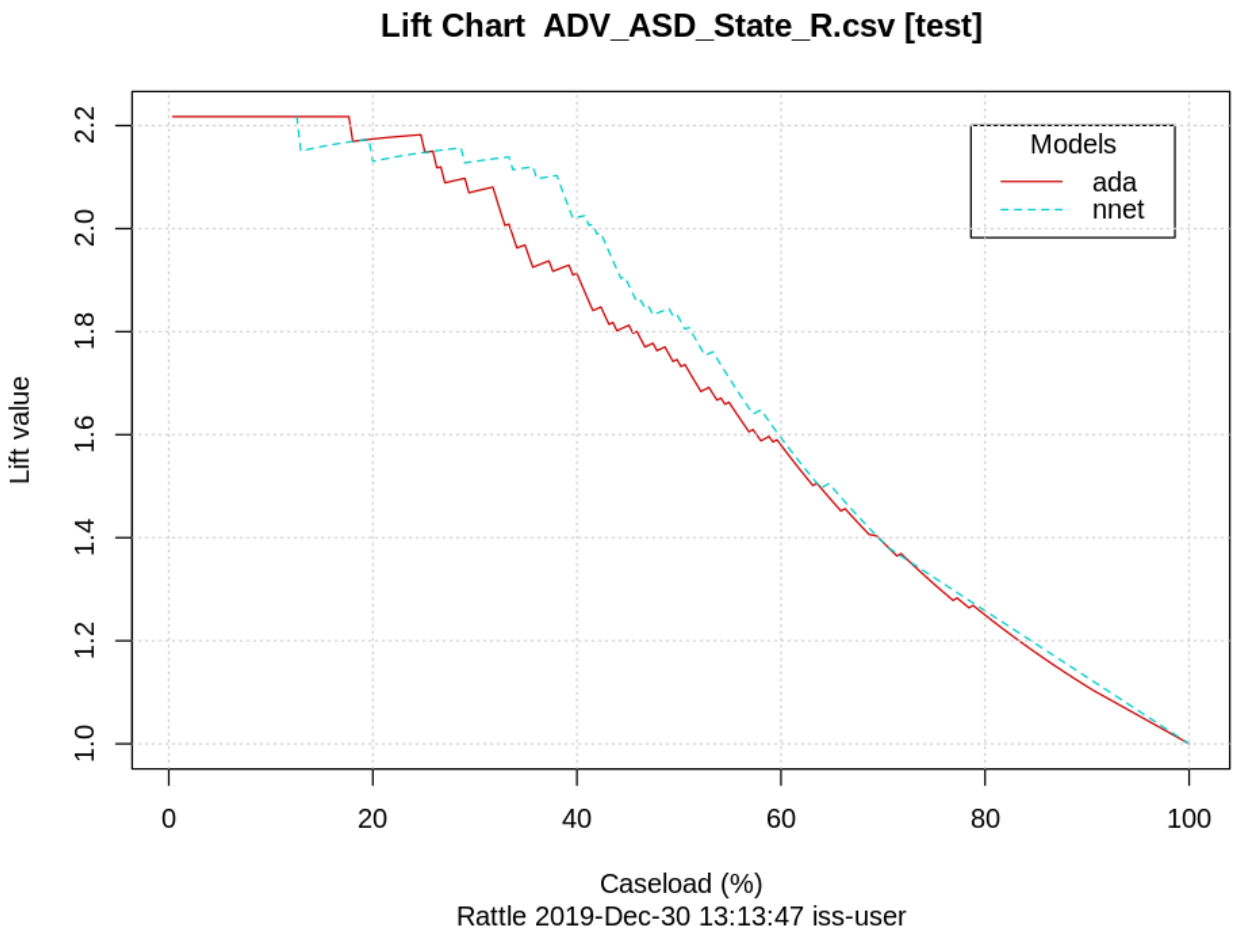
Rattle: Evaluate Model (Classification): Lift



In [73]:

```
#=====================================================================
# Rattle timestamp: 2019-12-24 12:11:55 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# Lift Chart: requires the ROCR package.

library(ROCR)

# Obtain predictions for the ada model on ADV_ASD_State_R.csv [test].

crs$pr <- predict(crs$ada, newdata=crs$dataset[crs$test, c(crs$input, crs$targ

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}

# Convert rate of positive predictions to percentage.

per <- performance(pred, "lift", "rpp")
per@x.values[[1]] <- per@x.values[[1]]*100

# Plot the lift chart.
ROCR::plot(per, col="#CC0000FF", lty=1, xlab="Caseload (%)", add=FALSE)

# Lift Chart: requires the ROCR package.

library(ROCR)

# Obtain predictions for the nnet model on ADV_ASD_State_R.csv [test].

crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input, crs$tar

# Remove observations with missing target.

no.miss   <- na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]$Prevalenc
miss.list <- attr(no.miss, "na.action")
attributes(no.miss) <- NULL

if (length(miss.list))
{
  pred <- prediction(crs$pr[-miss.list], no.miss)
} else
{
  pred <- prediction(crs$pr, no.miss)
}

# Convert rate of positive predictions to percentage.

per <- performance(pred, "lift", "rpp")
per@x.values[[1]] <- per@x.values[[1]]*100

# Plot the lift chart.
```

```
ROCR::plot(per, col="#00CCCCFF", lty=2, xlab="Caseload (%)", add=TRUE)

# Add a legend to the plot.

legend("topright", c("ada","nnet"), col=rainbow(2, 1, .8), lty=1:2, title="Mod

# Add decorations to the plot.

title(main="Lift Chart  ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()
```



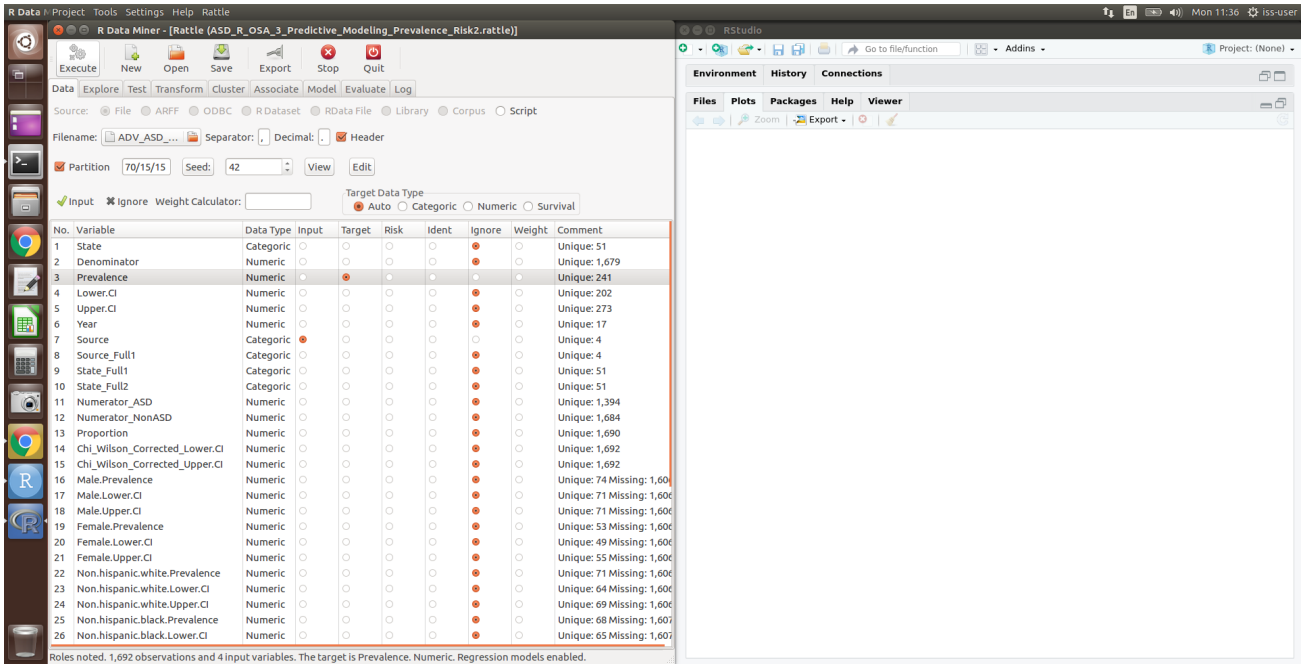
Rattle: Evaluate Model (Classification): Score/Write predicted results to CSV file.

In [74]:

```
#=======================================================================
# Rattle timestamp: 2019-12-24 12:22:22 x86_64-pc-linux-gnu

# Score the testing dataset.

# Obtain probability scores for the Extreme Boost model on ADV_ASD_State_R.csv

crs$pr_ada <- predict(crs$ada, newdata=crs$dataset[crs$test, c(crs$input)])

# Obtain probability scores for the Neural Net model on ADV_ASD_State_R.csv [t

crs$pr_nnet <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input)],

# Extract the relevant variables from the dataset.

sdata <- crs$dataset[crs$test,]

# Output the combined data.

write.csv(cbind(sdata, crs$pr_ada, crs$pr_nnet), file="../reference/R rattle/A
```

()

Rattle: Train Model (Regression)



In [75]:

```
#============================================================================
# Rattle timestamp: 2019-12-30 11:36:06 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# Build the train/validate/test datasets.

# nobs=1692 train=1184 validate=254 test=254

set.seed(crv$seed)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test

# The following variable selections have been noted.

crs$input     <- c("Source", "State_Region", "RMD_Year",
                   "RMD_R10_Denominator")

crs$numeric   <- c("RMD_Year", "RMD_R10_Denominator")

crs$categoric <- c("Source", "State_Region")

crs$target    <- "Prevalence"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- c("State", "Denominator", "Lower.CI", "Upper.CI", "Year", "So
crs$weights   <- NULL
```

In [76]:

```
crs$target
```

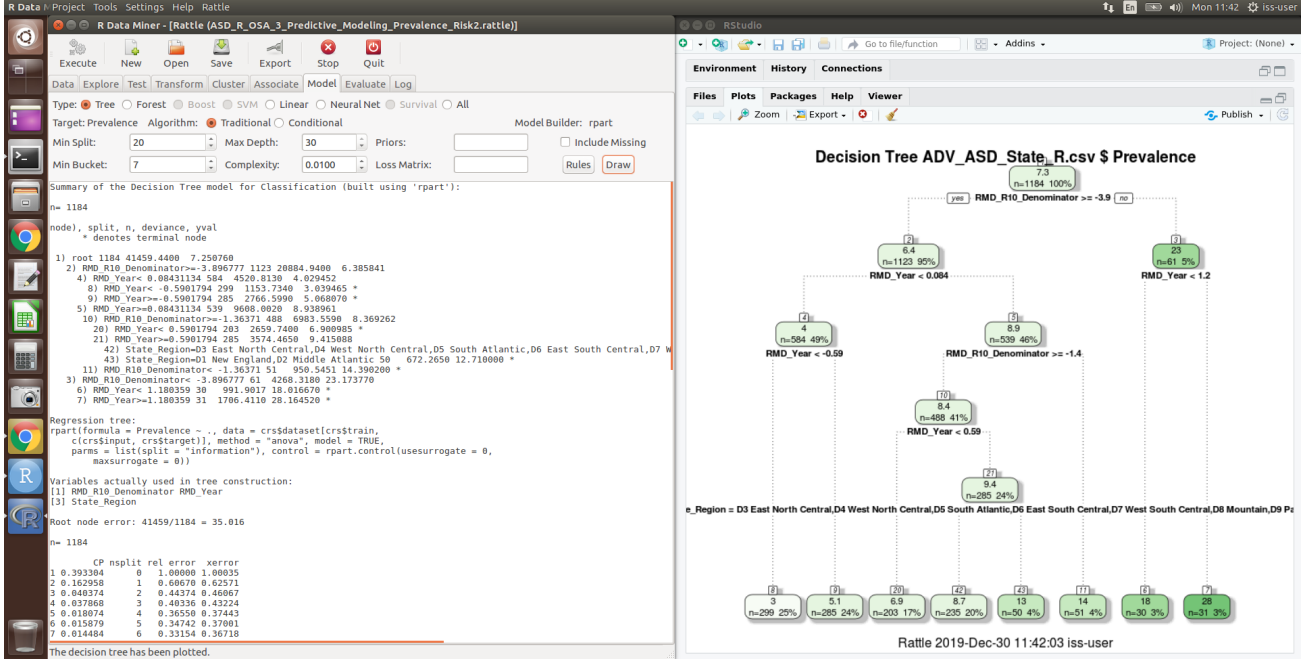
'Prevalence'

In [77]:

```
crs$input
```

'Source' 'State_Region' 'RMD_Year' 'RMD_R10_Denominator'

Regression Model: Decision Tree (DT)


R Data Miner - [Rattle (ASD_R_OSA_3_Predictive_Modeling_Prevalence_Risk2.rattle)]
Execute
New
Open
Save
Export
Stop
Quit
Data
Explore
Test
Transform
Cluster
Associate
Model
Evaluate
Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Target: Prevalence Algorithm: Traditional Conditional
Model Builder: rpart
Min Split: 20
Max Depth: 30
Priors:
Include Missing
Min Bucket: 7
Complexity: 0.0100
Loss Matrix:
Rules
Draw
Summary of the Decision Tree model for Classification (built using 'rpart'):
n= 1184
node), split, n, deviance, yval
* denotes terminal node
1) root 1184 41459.4400 7.250760
2) RMD_R10_Denominator>=-3.896777 1123 20884.9400 6.385841
4) RMD_Year< 0.08431134 584 4520.8130 4.029452
8) RMD_Year< -0.5901794 299 1153.7340 3.039465 *
9) RMD_Year>=-0.5901794 285 2766.5990 5.068070 *
5) RMD_Year>=0.08431134 539 9608.0020 8.938961
10) RMD_R10_Denominator>=-1.36371 488 6983.5590 8.369262
20) RMD_Year< 0.5901794 203 2659.7400 6.900985 *
21) RMD_Year>=0.5901794 285 3574.4650 9.415088
42) State_Region=D3 East North Central,D4 West North Central,D5 South Atlantic,D6 East South Central,D7 W
43) State_Region=D1 New England,D2 Middle Atlantic 50 672.2650 12.710000 *
11) RMD_R10_Denominator< -1.36371 51 950.5451 14.390200 *
3) RMD_R10_Denominator< -3.896777 61 4268.3180 23.173770
6) RMD_Year< 1.180359 30 991.9017 18.016670 *
7) RMD_Year>=1.180359 31 1706.4110 28.164520 *
Regression tree:
rpart(formula = Prevalence ~ ., data = crs$dataset[crs$train,
c(crs$input, crs$target)], method = "anova", model = TRUE,
parms = list(split = "information"), control = rpart.control(usesurrogate = 0,
maxsurrogate = 0))
Variables actually used in tree construction:
[1] RMD_R10_Denominator RMD_Year
[3] State_Region
Root node error: 41459/1184 = 35.016
n= 1184
CP nsplit rel error xerror
1 0.393304 0 1.00000 1.00035
2 0.162958 1 0.60670 0.62571
3 0.040374 2 0.44374 0.46067
4 0.037868 3 0.40336 0.43224
5 0.018074 4 0.36550 0.37443
6 0.015879 5 0.34742 0.37001
7 0.014484 6 0.33154 0.36718
The decision tree has been plotted.
RStudio
Environment
History
Connections
Files
Plots
Packages
Help
Viewer
Zoom
Export
Publish
Decision Tree ADV_ASD_State_R.csv $ Prevalence
7.3
n=1184 100%
yes
RMD_R10_Denominator >= -3.9
no
2
6.4
n=1123 95%
RMD_Year < 0.084
3
23
n=61 5%
RMD_Year < 1.2
4
4
n=584 49%
RMD_Year < -0.59
5
8.9
n=539 46%
RMD_R10_Denominator >= -1.4
10
8.4
n=488 41%
RMD_Year < 0.59
21
9.4
n=285 24%
e_Region = D3 East North Central,D4 West North Central,D5 South Atlantic,D6 East South Central,D7 West South Central,D8 Mountain,D9 Pa
8
3
n=299 25%
9
5.1
n=285 24%
20
6.9
n=203 17%
42
8.7
n=235 20%
43
13
n=50 4%
11
14
n=51 4%
6
18
n=30 3%
7
28
n=31 3%
Rattle 2019-Dec-30 11:42:03 iss-user

In [78]:

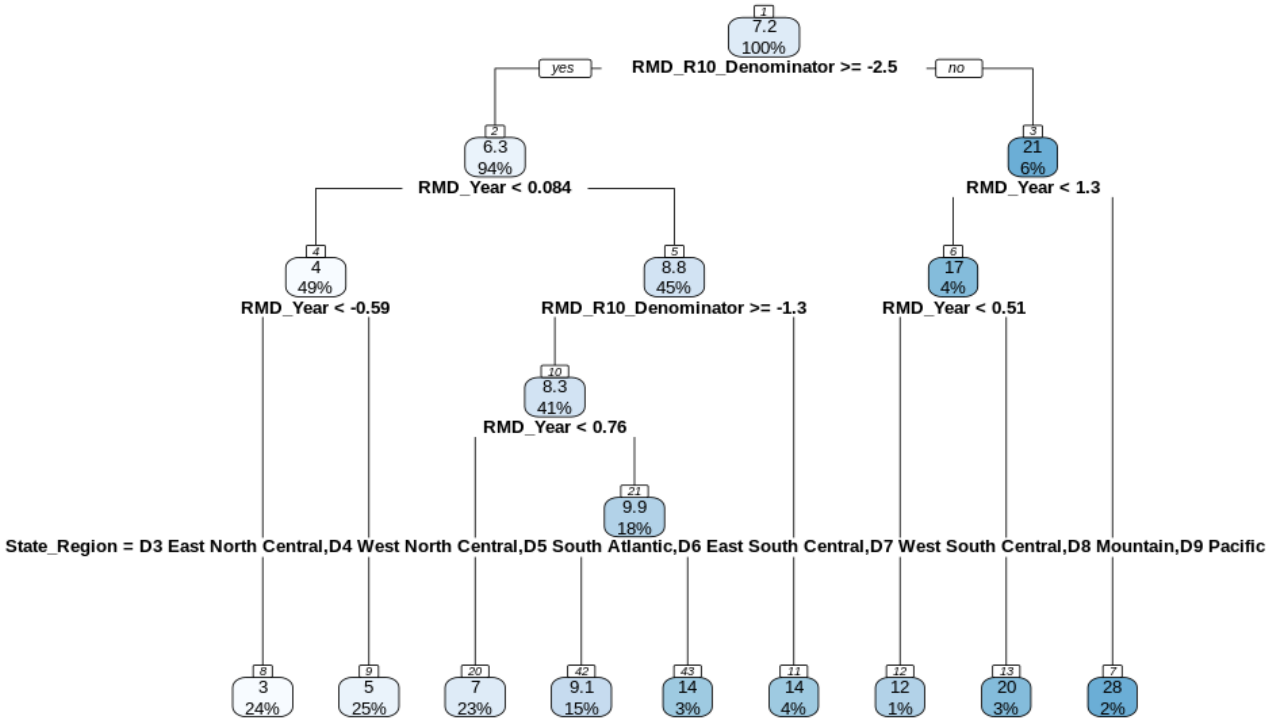
```
#=====================================================================
# Rattle timestamp: 2019-12-30 11:36:06 x86_64-pc-linux-gnu

# Action the user selections from the Data tab.

# Build the train/validate/test datasets.

# nobs=1692 train=1184 validate=254 test=254

set.seed(crv$seed)

crs$nobs <- nrow(crs$dataset)

crs$train <- sample(crs$nobs, 0.7*crs$nobs)

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  sample(0.15*crs$nobs) ->
crs$validate

crs$nobs %>%
  seq_len() %>%
  setdiff(crs$train) %>%
  setdiff(crs$validate) ->
crs$test

# The following variable selections have been noted.

crs$input     <- c("Source", "State_Region", "RMD_Year",
                   "RMD_R10_Denominator")

crs$numeric   <- c("RMD_Year", "RMD_R10_Denominator")

crs$categoric <- c("Source", "State_Region")

crs$target    <- "Prevalence"
crs$risk      <- NULL
crs$ident     <- NULL
crs$ignore    <- c("State", "Denominator", "Lower.CI", "Upper.CI", "Year", "So
crs$weights   <- NULL

#=====================================================================
# Rattle timestamp: 2019-12-30 11:41:57 x86_64-pc-linux-gnu

# Decision Tree

# The 'rpart' package provides the 'rpart' function.

library(rpart, quietly=TRUE)

# Reset the random number seed to obtain the same results each time.

set.seed(crv$seed)

# Build the Decision Tree model.

crs$rpart <- rpart(Prevalence ~ .,
    data=crs$dataset[crs$train, c(crs$input, crs$target)],
    method="anova",
    parms=list(split="information"),
    control=rpart.control(usesurrogate=0,
        maxsurrogate=0),
```

```
    model=TRUE)

# Generate a textual view of the Decision Tree model.

print(crs$rpart)
printcp(crs$rpart)
cat("\n")

# Time taken: 0.01 secs

# List the rules from the tree using a Rattle support function.

asRules(crs$rpart)
```

```
n= 1184

node), split, n, deviance, yval
      * denotes terminal node

 1) root 1184 38576.9200  7.213260
   2) RMD_R10_Denominator>=-2.544461 1111 19512.2000  6.315842
     4) RMD_Year< 0.08431134 576  4656.6530  4.046528
       8) RMD_Year< -0.5901794 283  1149.2920  3.036042 *
       9) RMD_Year>=-0.5901794 293  2939.2910  5.022526 *
     5) RMD_Year>=0.08431134 535  8695.6740  8.759065
      10) RMD_R10_Denominator>=-1.349522 491  6783.0690  8.309165
        20) RMD_Year< 0.7588021 276  3202.2730  7.038406 *
        21) RMD_Year>=0.7588021 215  2562.9580  9.940465
          42) State_Region=D3 East North Central,D4 West North Central,D5 S
outh Atlantic,D6 East South Central,D7 West South Central,D8 Mountain,D9 Pa
cific 176  1666.7780  9.128409 *
          43) State_Region=D1 New England,D2 Middle Atlantic 39   256.3590
13.605130 *
      11) RMD_R10_Denominator< -1.349522 44   704.1916 13.770550 *
```

In [79]:
```
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=6)
```

In [80]:
```
rpart.plot::prp(crs$rpart,
    type = 2, extra = "auto", nn = TRUE,
    under = FALSE, fallen.leaves = TRUE,
    digits = 2, varlen = 0, faclen = 0,
#    roundint = TRUE,
    cex = NULL, tweak = 1,
#    clip.facs = FALSE,
    clip.right.labs = TRUE,
    snip = FALSE,
    box.palette = "auto", shadow.col = 0)
```



Regression Model: Random Forest (RF)

R Data Miner Project Tools Settings Help Rattle

Mon 11:45 iss-user

R Data Miner - [Rattle (ASD_R_OSA_3_Predictive_Modeling_Prevalence_Risk2.rattle)]

Execute New Open Save Export Stop Quit

Data Explore Test Transform Cluster Associate Model Evaluate Log

Type: Tree Forest Boost SVM Linear Neural Net Survival All

Target: Prevalence Algorithm: Traditional Conditional Model Builder: randomForest

Trees: 500 Sample Size: Importance Rules 1

Variables: 2 Impute Errors OOB ROC

```
Summary of the Random Forest Model
==================================

Number of observations used to build the model: 1184
Missing value imputation is active.

Call:
 randomForest(formula = Prevalence ~ ., 
              data = crs$dataset[crs$train, c(crs$input, crs$target)], 
              ntree = 500, mtry = 2, importance = TRUE, replace = FALSE, na.action = randomForest::na.roughfix)

               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 8.514544
                    % Var explained: 75.68

Variable Importance
===================

                      %IncMSE IncNodePurity
RMD_Year               103.92       7165.74
State_Region            58.19       2206.20
RMD_R10_Denominator     44.27       9325.08
Source                  26.22       4730.90

Time taken: 1.17 secs

Rattle timestamp: 2019-12-30 11:45:06 iss-user
```

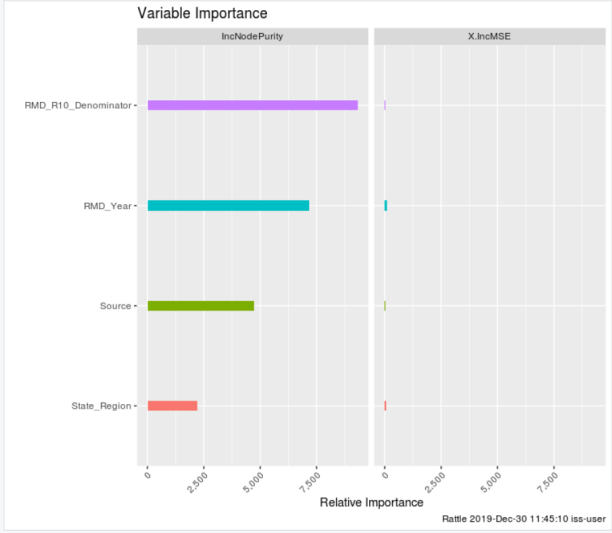
RStudio

Go to file/function Addins Project: (None)

Environment History Connections

Files Plots Packages Help Viewer

Zoom Export Publish

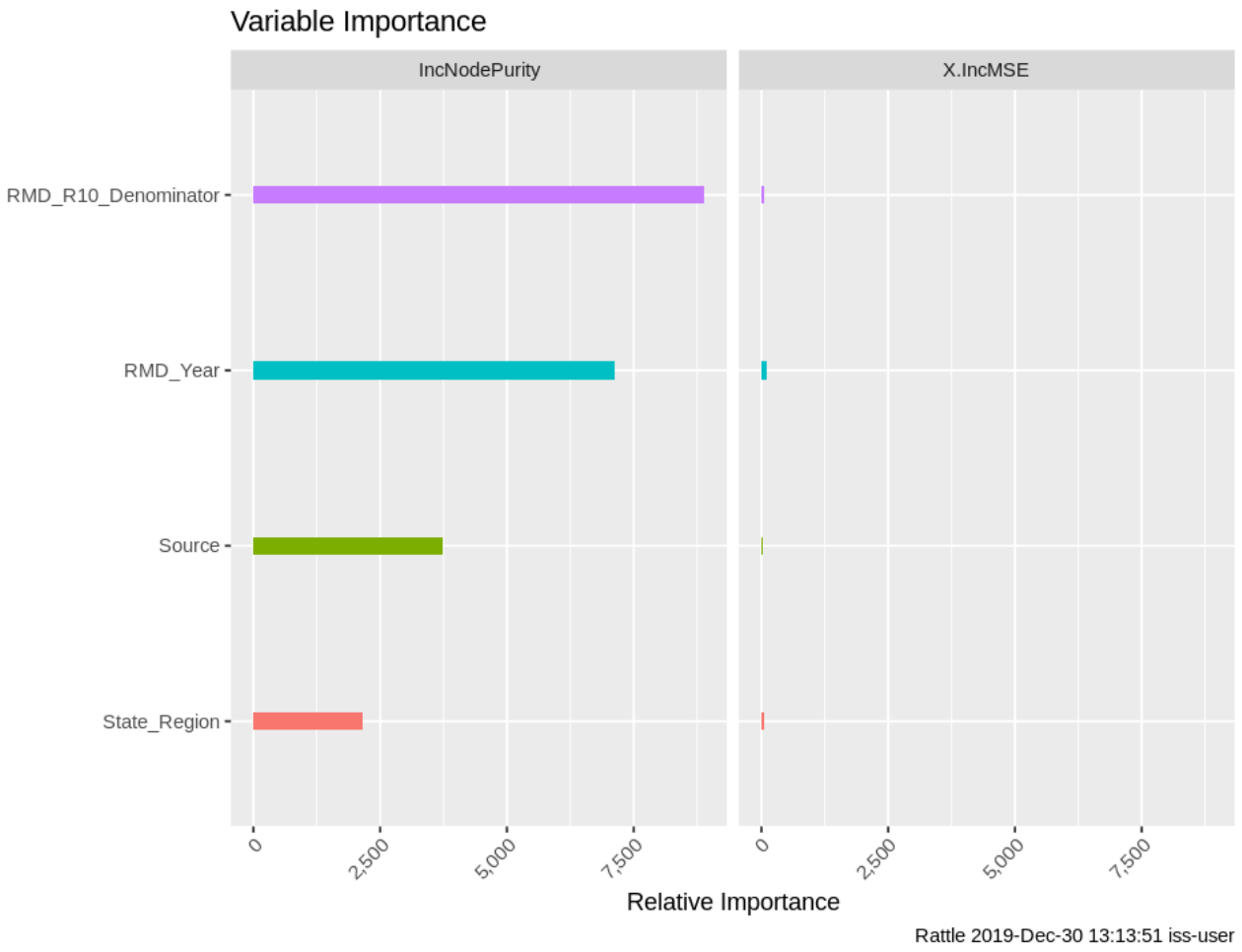


In [81]:

```
#=====================================================================
# Rattle timestamp: 2019-12-30 11:45:04 x86_64-pc-linux-gnu

# Build a Random Forest model using the traditional approach.

set.seed(crv$seed)

crs$rf <- randomForest::randomForest(Prevalence ~ .,
  data=crs$dataset[crs$train, c(crs$input, crs$target)],
  ntree=500,
  mtry=2,
  importance=TRUE,
  na.action=randomForest::na.roughfix,
  replace=FALSE)

# Generate textual output of the 'Random Forest' model.

crs$rf

# List the importance of the variables.

rn <- crs$rf %>%
    randomForest::importance() %>%
    round(2)
    rn[order(rn[,1], decreasing=TRUE),]

# Time taken: 1.17 secs

#=====================================================================
# Rattle timestamp: 2019-12-30 11:45:10 x86_64-pc-linux-gnu

# Plot the relative importance of the variables.

p <- ggVarImp(crs$rf,
              title="Variable Importance Random Forest ADV_ASD_State_R.csv")
p
```

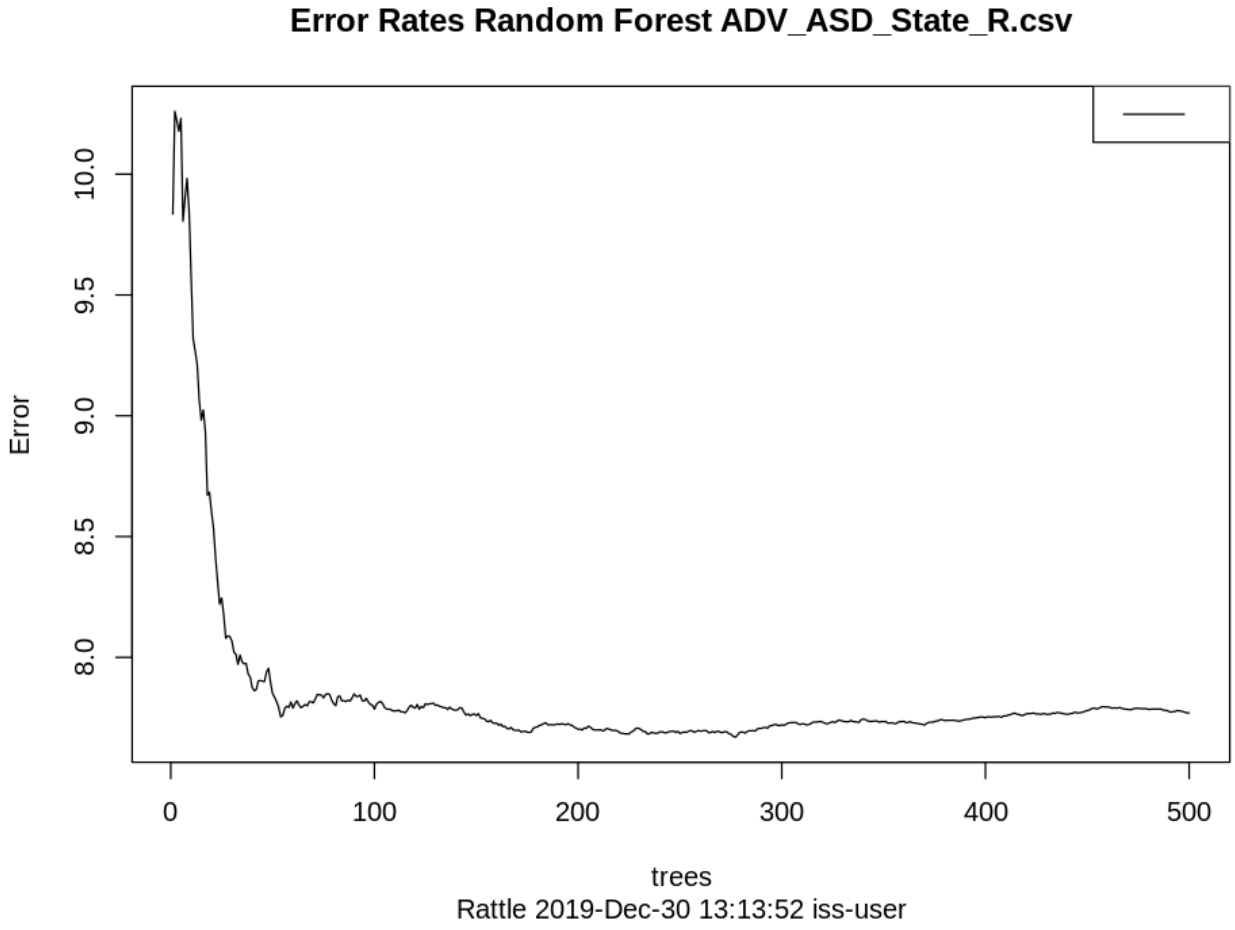
```
Call:
 randomForest(formula = Prevalence ~ ., data = crs$dataset[crs$train,      c
(crs$input, crs$target)], ntree = 500, mtry = 2, importance = TRUE,      repl
ace = FALSE, na.action = randomForest::na.roughfix)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 2

          Mean of squared residuals: 7.770573
                    % Var explained: 76.15
```

	%IncMSE	IncNodePurity
RMD_Year	114.09	7137.03
State_Region	58.32	2156.68
RMD_R10_Denominator	43.02	8889.71
Source	25.21	3722.86


Variable Importance
IncNodePurity
X.IncMSE
RMD_R10_Denominator
RMD_Year
Source
State_Region
0
2,500
5,000
7,500
0
2,500
5,000
7,500
Relative Importance
Rattle 2019-Dec-30 13:13:51 iss-user

In [82]:

```
# Plot the error rate against the number of trees.

plot(crs$rf, main="")
legend("topright", c(""), text.col=1:6, lty=1:3, col=1:3)
title(main="Error Rates Random Forest ADV_ASD_State_R.csv",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
```

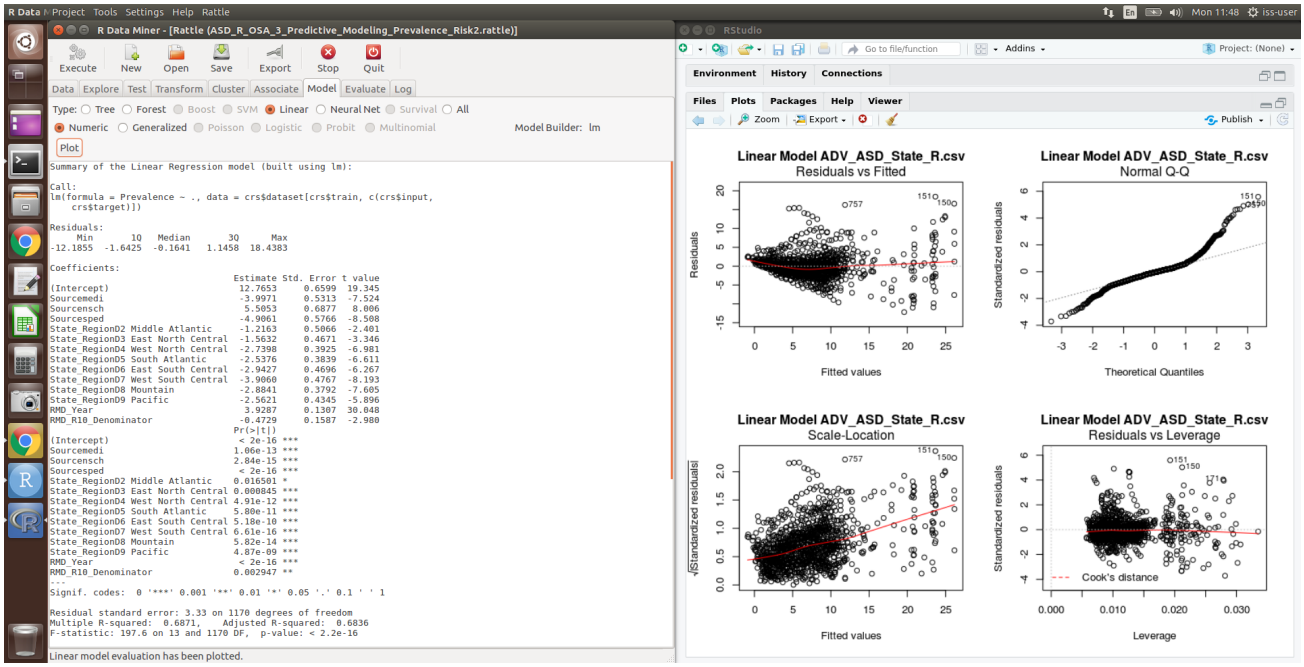


In [83]:

```
# Display tree number 1.

printRandomForests(crs$rf, 1)
```

```
Random Forest Model 1

------------------------------------------------------------
Tree 1 Rule 1 Node 8 Regression (to do - extract predicted value)

1: RMD_R10_Denominator <= -3.89236192492941
2: RMD_Year <= 0.505868069607446
3: State_Region IN ("D1 New England", "D2 Middle Atlantic", "D4 West North
Central", "D5 South Atlantic", "D6 East South Central", "D7 West South Cent
ral", "D8 Mountain")
------------------------------------------------------------
Tree 1 Rule 2 Node 9 Regression (to do - extract predicted value)

1: RMD_R10_Denominator <= -3.89236192492941
2: RMD_Year <= 0.505868069607446
3: State_Region IN ("D3 East North Central", "D9 Pacific")
------------------------------------------------------------
Tree 1 Rule 3 Node 16 Regression (to do - extract predicted value)
```

Regression Model: Linear Regression (LR)

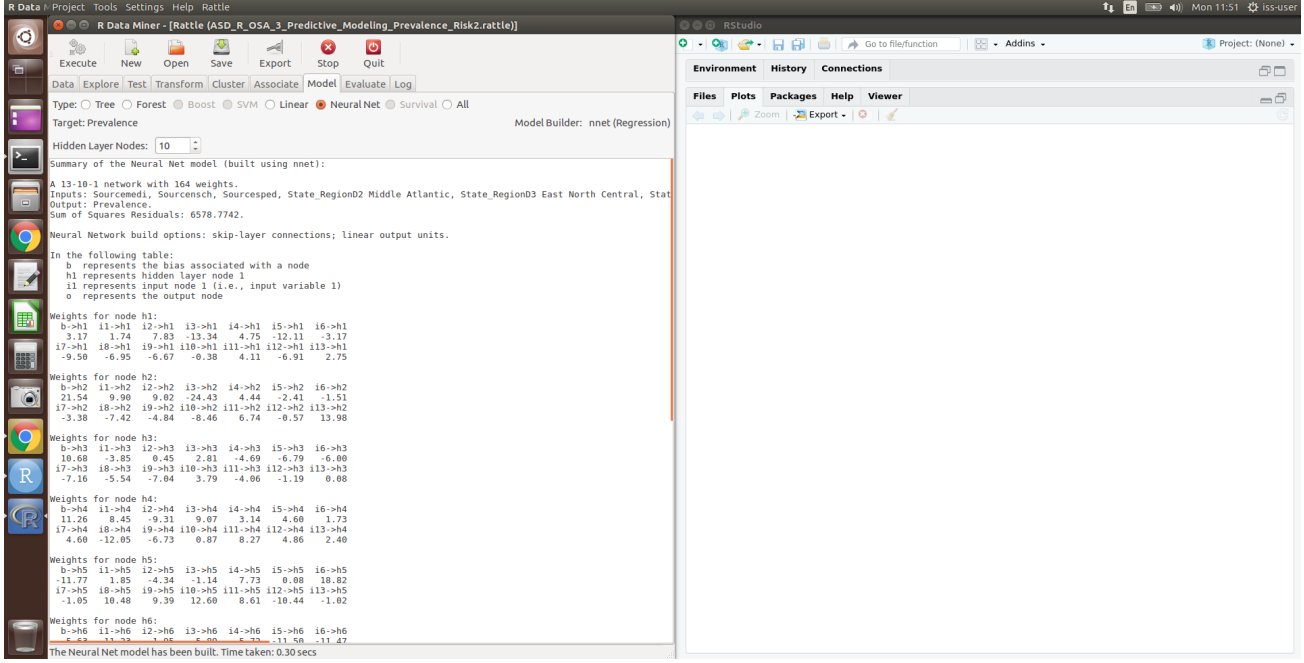


In [84]:

```
#============================================================================
# Rattle timestamp: 2019-12-30 11:48:44 x86_64-pc-linux-gnu

# Regression model

# Build a Regression model.

crs$glm <- lm(Prevalence ~ ., data=crs$dataset[crs$train,c(crs$input, crs$targ

# Generate a textual view of the Linear model.

print(summary(crs$glm))
cat('==== ANOVA ====

')
print(anova(crs$glm))
print("
")

# Time taken: 0.28 secs

# Plot the model evaluation.

ttl <- genPlotTitleCmd("Linear Model",crs$dataname,vector=TRUE)
plot(crs$glm, main=ttl[1])
```

```
Call:
lm(formula = Prevalence ~ ., data = crs$dataset[crs$train, c(crs$input,
    crs$target)])

Residuals:
     Min       1Q   Median       3Q      Max
-11.4991  -1.6532  -0.1294   1.0765  15.7987

Coefficients:
                                    Estimate Std. Error t value Pr(>|t|)
(Intercept)                          12.8301     0.6337  20.247  < 2e-16 ***
Sourcemedi                           -4.0030     0.5122  -7.816 1.21e-14 ***
Sourcensch                            5.3449     0.6595   8.105 1.32e-15 ***
Sourcesped                           -4.8738     0.5521  -8.827  < 2e-16 ***
State_RegionD2 Middle Atlantic       -1.4687     0.4947  -2.969  0.00305 **
State_RegionD3 East North Central    -1.7105     0.4408  -3.881  0.00011 ***
State_RegionD4 West North Central    -2.2716     0.3790  -5.994 2.72e-09 ***
State_RegionD5 South Atlantic        -2.7806     0.3663  -7.592 6.44e-14 ***
State_RegionD6 East South Central    -3.2525     0.4513  -7.428 2.12e-13 ***
```

Regression Model: Neural Network (NN)


R Data Project Tools Settings Help Rattle
Mon 11:51 iss-user
R Data Miner - [Rattle (ASD_R_OSA_3_Predictive_Modeling_Prevalence_Risk2.rattle)]
Execute New Open Save Export Stop Quit
Data Explore Test Transform Cluster Associate Model Evaluate Log
Type: Tree Forest Boost SVM Linear Neural Net Survival All
Target: Prevalence
Model Builder: nnet (Regression)
Hidden Layer Nodes: 10
Summary of the Neural Net model (built using nnet):
A 13-10-1 network with 164 weights.
Inputs: Sourcemedi, Sourcensch, Sourcesped, State_RegionD2 Middle Atlantic, State_RegionD3 East North Central, Stat
Output: Prevalence.
Sum of Squares Residuals: 6578.7742.
Neural Network build options: skip-layer connections; linear output units.
In the following table:
b represents the bias associated with a node
h1 represents hidden layer node 1
i1 represents input node 1 (i.e., input variable 1)
o represents the output node
Weights for node h1:
b->h1 i1->h1 i2->h1 i3->h1 i4->h1 i5->h1 i6->h1
3.17 1.74 7.83 -13.34 4.75 -12.11 -3.17
i7->h1 i8->h1 i9->h1 i10->h1 i11->h1 i12->h1 i13->h1
-9.50 -6.95 -6.67 -0.38 4.11 -6.91 2.75
Weights for node h2:
b->h2 i1->h2 i2->h2 i3->h2 i4->h2 i5->h2 i6->h2
21.54 9.90 9.02 -24.43 4.44 -2.41 -1.51
i7->h2 i8->h2 i9->h2 i10->h2 i11->h2 i12->h2 i13->h2
-3.38 -7.42 -4.84 -8.46 6.74 -0.57 13.98
Weights for node h3:
b->h3 i1->h3 i2->h3 i3->h3 i4->h3 i5->h3 i6->h3
10.68 -3.85 0.45 2.81 -4.69 -6.79 -6.00
i7->h3 i8->h3 i9->h3 i10->h3 i11->h3 i12->h3 i13->h3
-7.16 -5.54 -7.04 3.79 -4.06 -1.19 0.08
Weights for node h4:
b->h4 i1->h4 i2->h4 i3->h4 i4->h4 i5->h4 i6->h4
11.26 8.45 -9.31 9.07 3.14 4.60 1.73
i7->h4 i8->h4 i9->h4 i10->h4 i11->h4 i12->h4 i13->h4
4.60 -12.05 -6.73 0.87 8.27 4.86 2.40
Weights for node h5:
b->h5 i1->h5 i2->h5 i3->h5 i4->h5 i5->h5 i6->h5
-11.77 1.85 -4.34 -1.14 7.73 0.08 18.82
i7->h5 i8->h5 i9->h5 i10->h5 i11->h5 i12->h5 i13->h5
-1.05 10.48 9.39 12.60 8.61 -10.44 -1.02
Weights for node h6:
b->h6 i1->h6 i2->h6 i3->h6 i4->h6 i5->h6 i6->h6
The Neural Net model has been built. Time taken: 0.30 secs
RStudio
Go to file/function
Addins
Project: (None)
Environment History Connections
Files Plots Packages Help Viewer
Zoom
Export

In [85]:

```
#============================================================================
# Rattle timestamp: 2019-12-30 11:51:47 x86_64-pc-linux-gnu

# Neural Network

# Build a neural network model using the nnet package.

library(nnet, quietly=TRUE)

# Build the NNet model.

set.seed(199)
crs$nnet <- nnet(Prevalence ~ .,
    data=crs$dataset[crs$train,c(crs$input, crs$target)],
    size=10, linout=TRUE, skip=TRUE, MaxNWts=10000, trace=FALSE, maxit=100)

# Print the results of the modelling.

cat(sprintf("A %s network with %d weights.\n",
    paste(crs$nnet$n, collapse="-"),
    length(crs$nnet$wts)))
cat(sprintf("Inputs: %s.\n",
    paste(crs$nnet$coefnames, collapse=", ")))
cat(sprintf("Output: %s.\n",
    names(attr(crs$nnet$terms, "dataClasses"))[1]))
cat(sprintf("Sum of Squares Residuals: %.4f.\n",
    sum(residuals(crs$nnet) ^ 2)))
cat("\n")
print(summary(crs$nnet))
cat('\n')

# Time taken: 0.30 secs
```

```
A 13-10-1 network with 164 weights.
Inputs: Sourcemedi, Sourcensch, Sourcesped, State_RegionD2 Middle Atlantic,
State_RegionD3 East North Central, State_RegionD4 West North Central, State
_RegionD5 South Atlantic, State_RegionD6 East South Central, State_RegionD7
West South Central, State_RegionD8 Mountain, State_RegionD9 Pacific, RMD_Ye
ar, RMD_R10_Denominator.
Output: Prevalence.
Sum of Squares Residuals: 5961.7621.

a 13-10-1 network with 164 weights
options were - skip-layer connections  linear output units
  b->h1  i1->h1  i2->h1  i3->h1  i4->h1  i5->h1  i6->h1  i7->h1  i8->h1  i9
->h1
  16.82  -14.75    2.09  -15.74  -13.74   -8.97   -8.67   -9.77   -8.20   -
8.01
i10->h1 i11->h1 i12->h1 i13->h1
  -4.46   -8.71   -5.51    0.88
  b->h2  i1->h2  i2->h2  i3->h2  i4->h2  i5->h2  i6->h2  i7->h2  i8->h2  i9
->h2
```

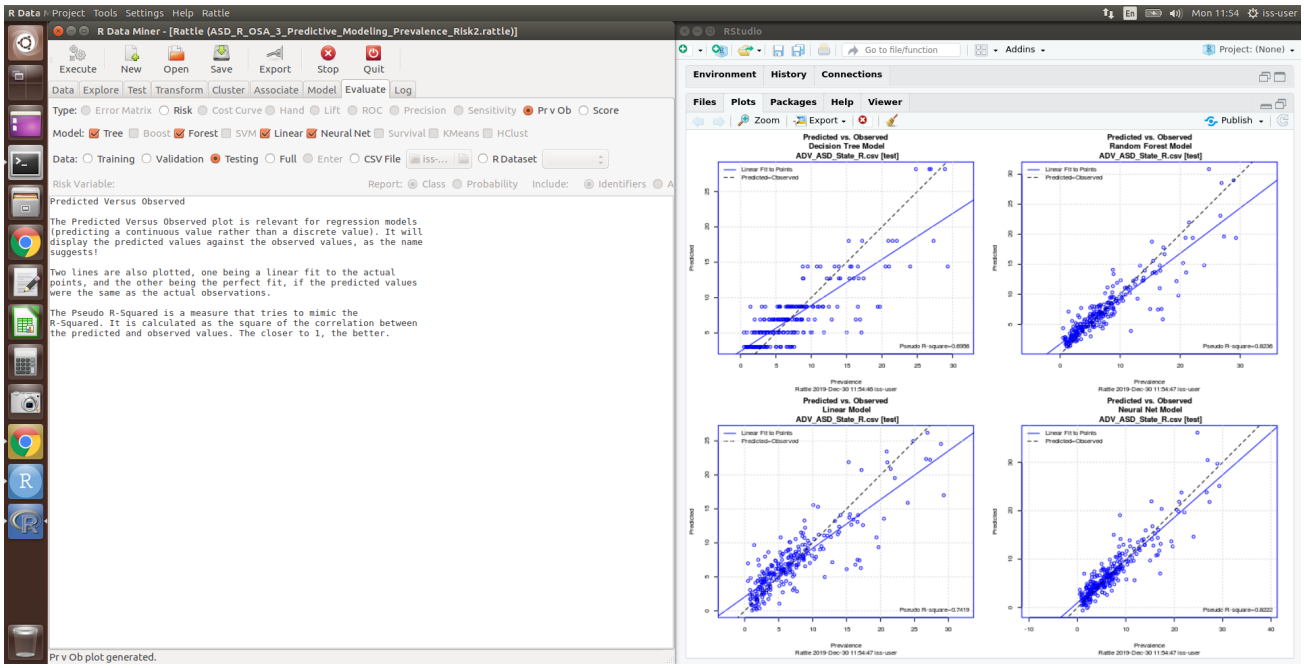
Rattle: Evaluate Model (Regression)

Regression: Prevalence

```
crs$rpart
crs$rf
```

```
crs$glm
crs$nnet
```

Rattle: Evaluate Model (Regression): Predicted Versus Observed



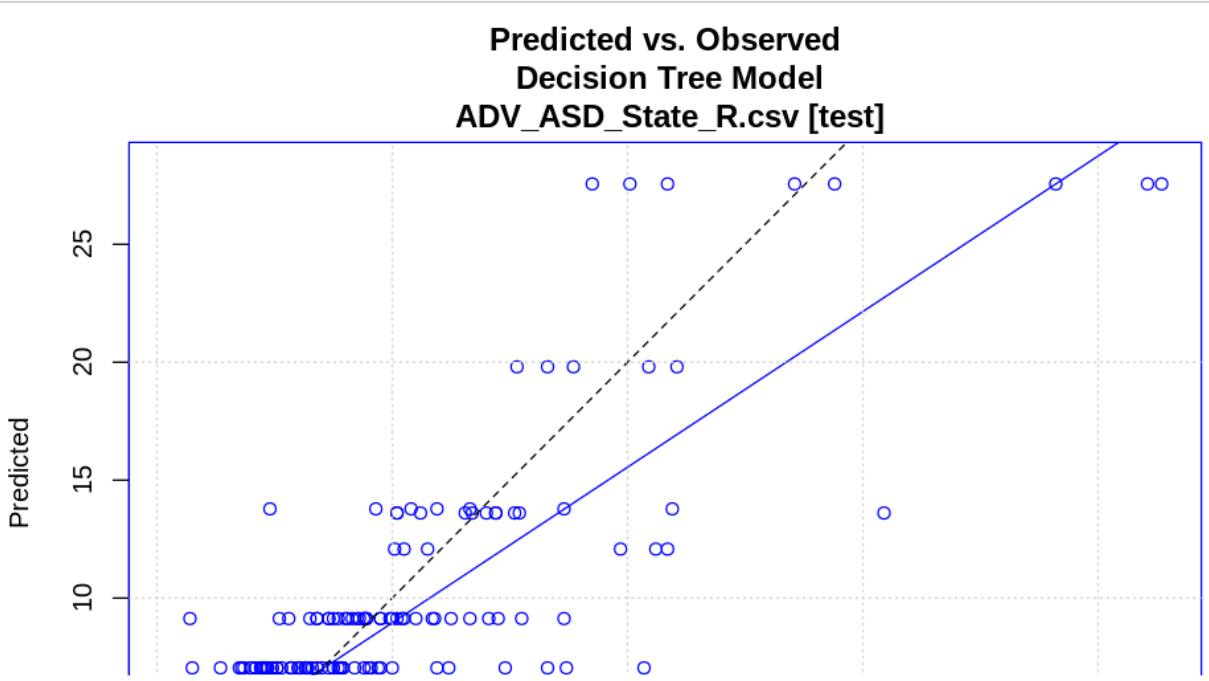
In [86]:

```
#=======================================================================
# Rattle timestamp: 2019-12-30 11:54:46 x86_64-pc-linux-gnu

# Evaluate model performance on the testing dataset.

# RPART: Generate a Predicted v Observed plot for rpart model on ADV_ASD_State

crs$pr <- predict(crs$rpart, newdata=crs$dataset[crs$test, c(crs$input, crs$ta

# Obtain the observed output for the dataset.

obs <- subset(crs$dataset[crs$test, c(crs$input, crs$target)], select=crs$targ

# Handle in case categoric target treated as numeric.

obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(Prevalence=obs)
rownames(obs) <- obs.rownames

# Combine the observed values with the predicted.

fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))

# Obtain the pseudo R2 - a correlation.

fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)

# Plot settings for the true points and best fit.

op <- par(c(lty="solid", col="blue"))

# Display the observed (X) versus predicted (Y) points.

plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="Prevalence", ylab="Predicted

# Generate a simple linear fit between predicted and observed.

prline <- lm(fitpoints[,2] ~ fitpoints[,1])

# Add the linear fit to the plot.

abline(prline)

# Add a diagonal representing perfect correlation.

par(c(lty="dashed", col="black"))
abline(0, 1)

# Include a pseudo R-square on the plot

legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")

# Add a title and grid to the plot.

title(main="Predicted vs. Observed
 Decision Tree Model
 ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()

# RF: Generate a Predicted v Observed plot for rf model on ADV_ASD_State_R.csv
```

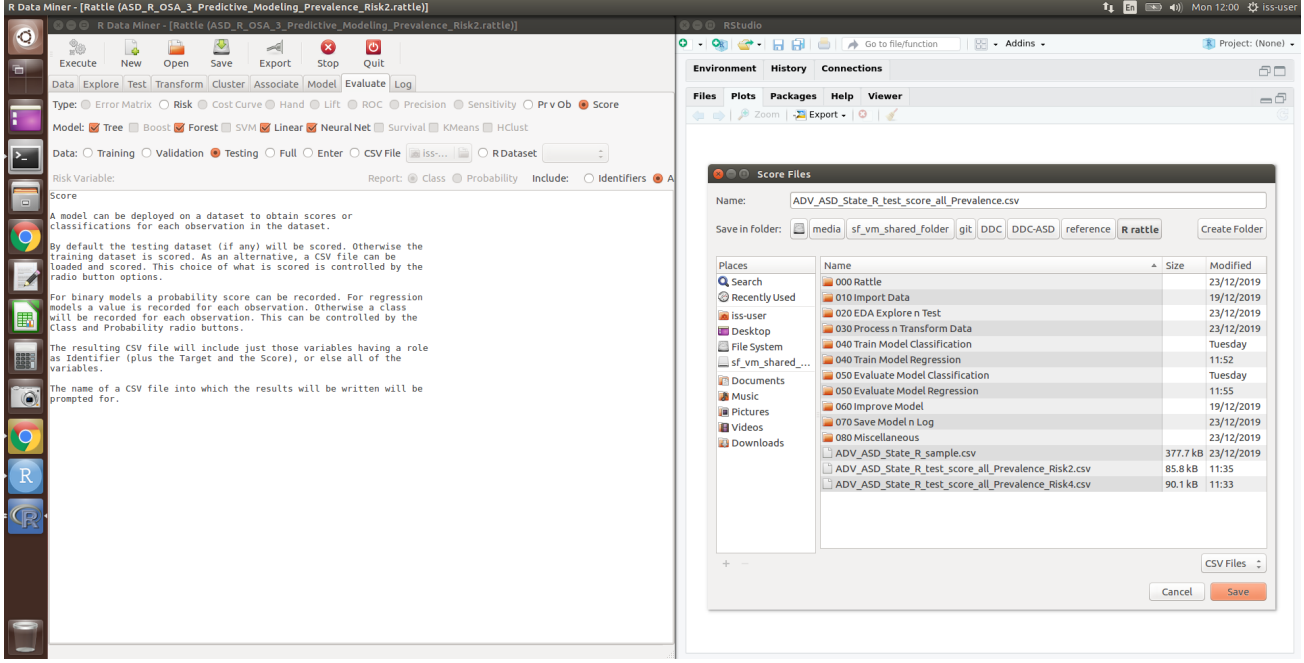
```
crs$pr <- predict(crs$rf, newdata=na.omit(crs$dataset[crs$test, c(crs$input, c

# Obtain the observed output for the dataset.

obs <- subset(na.omit(crs$dataset[crs$test, c(crs$input, crs$target)]), select

# Handle in case categoric target treated as numeric.

obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(Prevalence=obs)
rownames(obs) <- obs.rownames

# Combine the observed values with the predicted.

fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))

# Obtain the pseudo R2 - a correlation.

fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)

# Plot settings for the true points and best fit.

op <- par(c(lty="solid", col="blue"))

# Display the observed (X) versus predicted (Y) points.

plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="Prevalence", ylab="Predicted

# Generate a simple linear fit between predicted and observed.

prline <- lm(fitpoints[,2] ~ fitpoints[,1])

# Add the linear fit to the plot.

abline(prline)

# Add a diagonal representing perfect correlation.

par(c(lty="dashed", col="black"))
abline(0, 1)

# Include a pseudo R-square on the plot

legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")

# Add a title and grid to the plot.

title(main="Predicted vs. Observed
 Random Forest Model
 ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()

# GLM: Generate a Predicted v Observed plot for glm model on ADV_ASD_State_R.c

crs$pr <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[crs$test, c(crs$input, crs$target)])

# Obtain the observed output for the dataset.

obs <- subset(crs$dataset[crs$test, c(crs$input, crs$target)], select=crs$targ

# Handle in case categoric target treated as numeric.
```

```
obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(Prevalence=obs)
rownames(obs) <- obs.rownames

# Combine the observed values with the predicted.

fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))

# Obtain the pseudo R2 - a correlation.

fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)

# Plot settings for the true points and best fit.

op <- par(c(lty="solid", col="blue"))

# Display the observed (X) versus predicted (Y) points.

plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="Prevalence", ylab="Predicted

# Generate a simple linear fit between predicted and observed.

prline <- lm(fitpoints[,2] ~ fitpoints[,1])

# Add the linear fit to the plot.

abline(prline)

# Add a diagonal representing perfect correlation.

par(c(lty="dashed", col="black"))
abline(0, 1)

# Include a pseudo R-square on the plot

legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")

# Add a title and grid to the plot.

title(main="Predicted vs. Observed
 Linear Model
 ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()

# NNET: Generate a Predicted v Observed plot for nnet model on ADV_ASD_State_R

crs$pr <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input, crs$tar

# Obtain the observed output for the dataset.

obs <- subset(crs$dataset[crs$test, c(crs$input, crs$target)], select=crs$targ

# Handle in case categoric target treated as numeric.

obs.rownames <- rownames(obs)
obs <- as.numeric(obs[[1]])
obs <- data.frame(Prevalence=obs)
rownames(obs) <- obs.rownames

# Combine the observed values with the predicted.

fitpoints <- na.omit(cbind(obs, Predicted=crs$pr))

# Obtain the pseudo R2 - a correlation.
```

```
fitcorr <- format(cor(fitpoints[,1], fitpoints[,2])^2, digits=4)

# Plot settings for the true points and best fit.

op <- par(c(lty="solid", col="blue"))

# Display the observed (X) versus predicted (Y) points.

plot(fitpoints[[1]], fitpoints[[2]], asp=1, xlab="Prevalence", ylab="Predicted

# Generate a simple linear fit between predicted and observed.

prline <- lm(fitpoints[,2] ~ fitpoints[,1])

# Add the linear fit to the plot.

abline(prline)

# Add a diagonal representing perfect correlation.

par(c(lty="dashed", col="black"))
abline(0, 1)

# Include a pseudo R-square on the plot

legend("bottomright",  sprintf(" Pseudo R-square=%s ", fitcorr),  bty="n")

# Add a title and grid to the plot.

title(main="Predicted vs. Observed
 Neural Net Model
 ADV_ASD_State_R.csv [test]",
    sub=paste("Rattle", format(Sys.time(), "%Y-%b-%d %H:%M:%S"), Sys.info()["u
grid()
```



Rattle: Evaluate Model (Regression): Score/Write predicted results to CSV file.



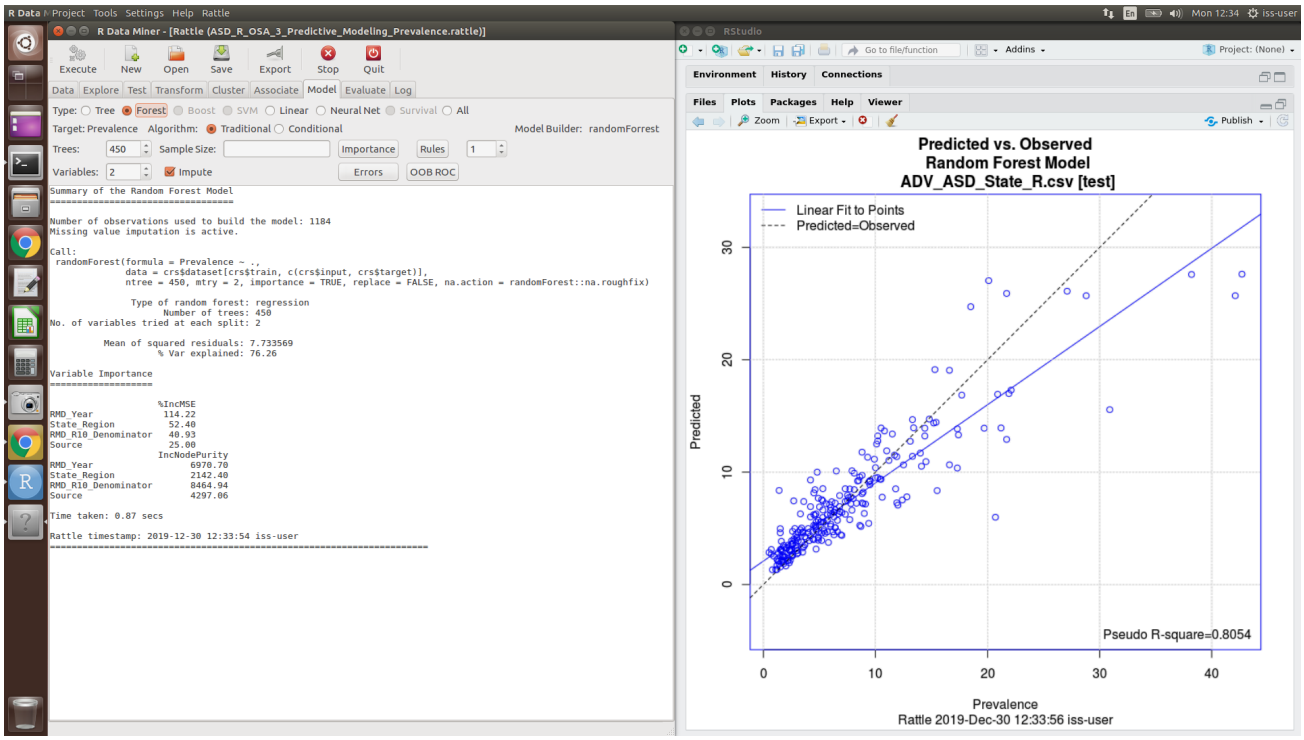
In [87]:

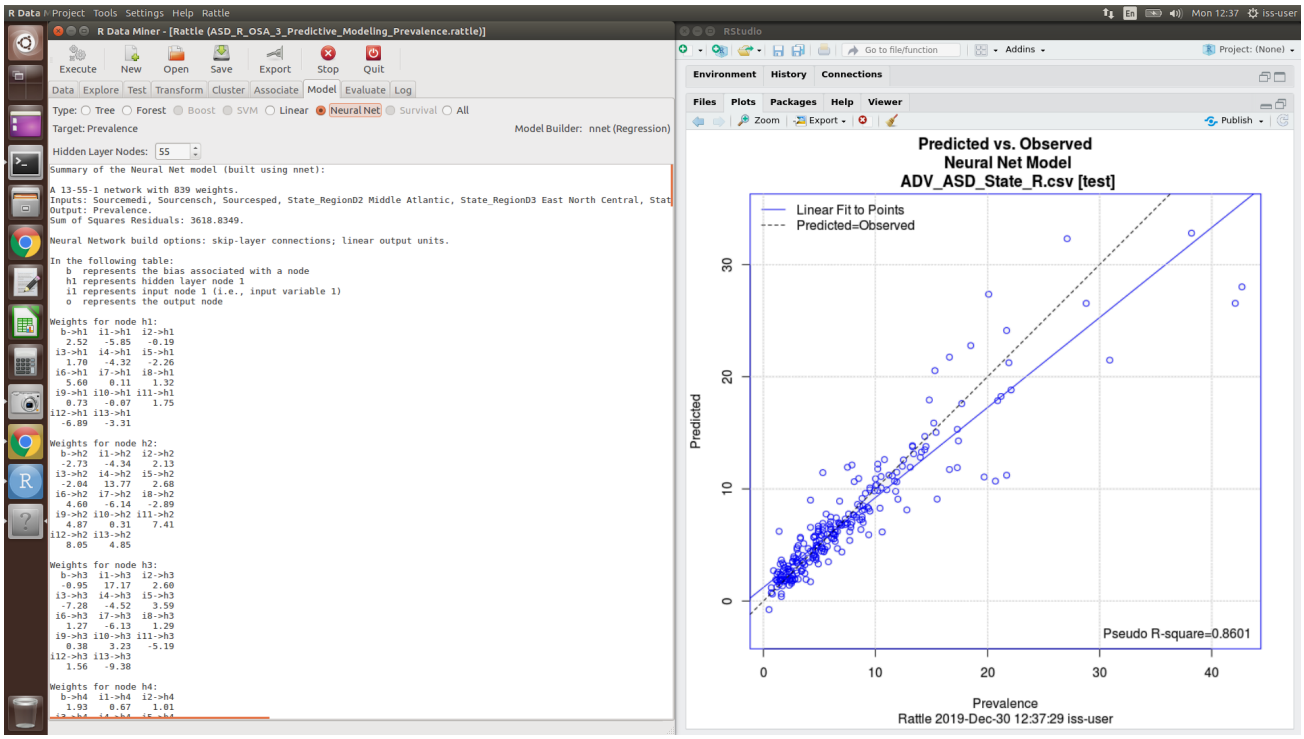
```
#=====================================================================
# Rattle timestamp: 2019-12-30 11:59:50 x86_64-pc-linux-gnu

# Score the testing dataset.

# Obtain predictions for the Decision Tree model on ADV_ASD_State_R.csv [test]

crs$pr_rpart <- predict(crs$rpart, newdata=crs$dataset[crs$test, c(crs$input)]

# Obtain predictions for the Random Forest model on ADV_ASD_State_R.csv [test]

crs$pr_rf <- predict(crs$rf, newdata=na.omit(crs$dataset[crs$test, c(crs$input

# Obtain predictions for the Linear model on ADV_ASD_State_R.csv [test].

crs$pr_glm <- predict(crs$glm,
   type    = "response",
   newdata = crs$dataset[crs$test, c(crs$input)])

# Obtain predictions for the Neural Net model on ADV_ASD_State_R.csv [test].

crs$pr_nnet <- predict(crs$nnet, newdata=crs$dataset[crs$test, c(crs$input)])

# Extract the relevant variables from the dataset.

sdata <- crs$dataset[crs$test,]

# Output the combined data.

write.csv(cbind(sdata, crs$pr_rpart, crs$pr_rf, crs$pr_glm, crs$pr_nnet), file
```

()

Rattle: Improve Model

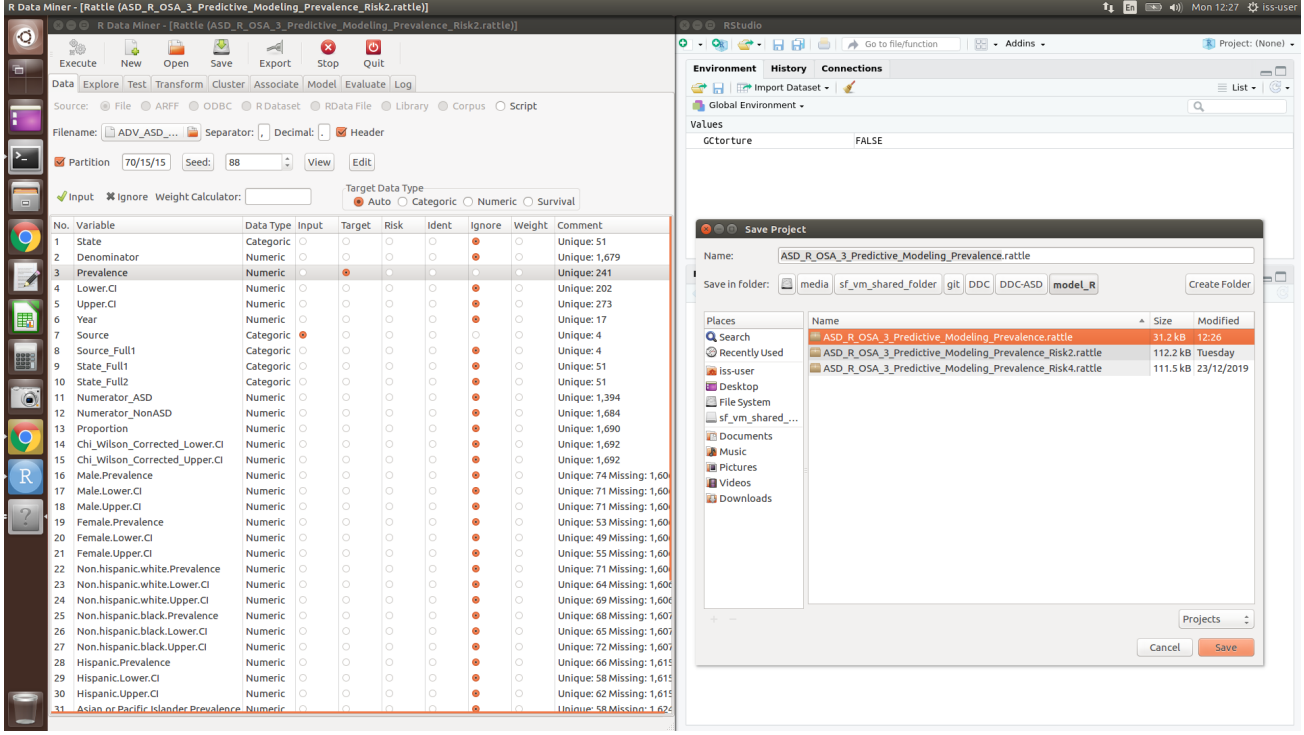
Tune hyperparameters

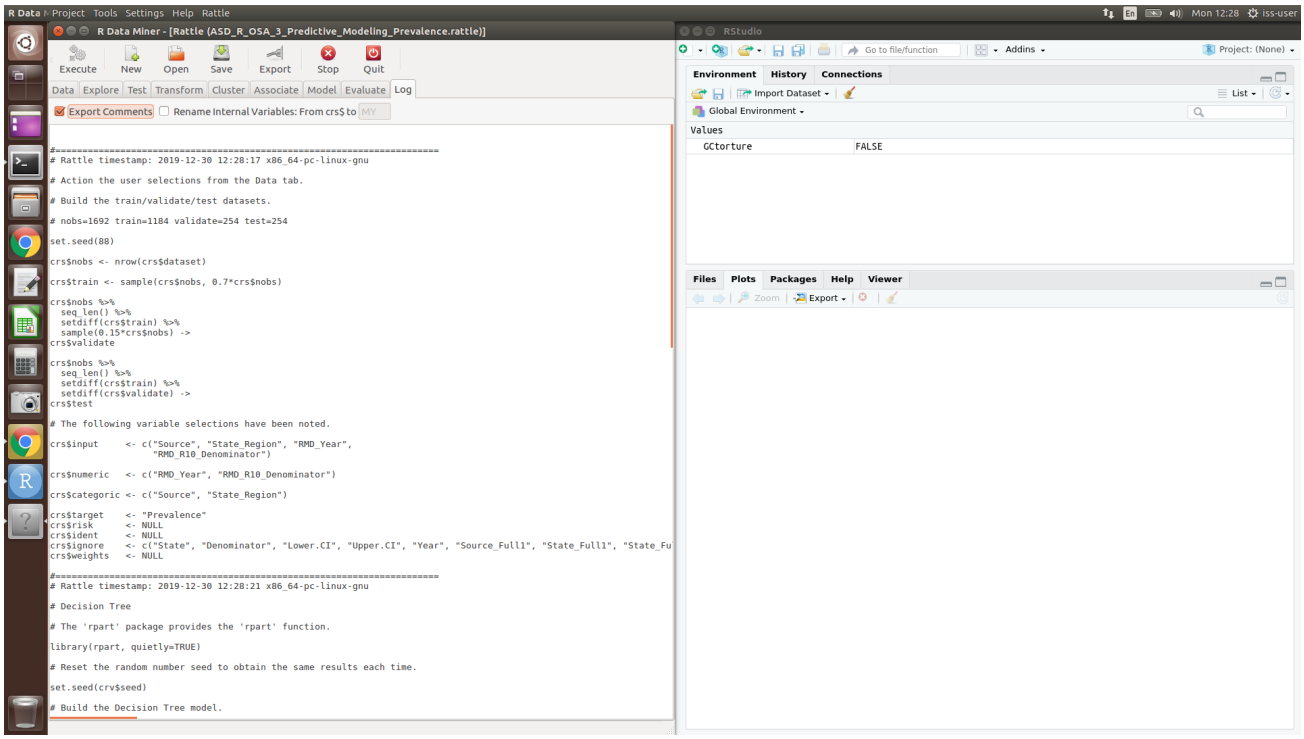




()

Rattle: Save Model & Log





()

Workshop Submission

What to submit?

Create predictive model for Multi Class Classification of ASD Prevalence Risk Level (Low, Medium, High, Very High) using xgboost package's Extreme Gradient Boosting algorithm.

References:

XGBoost support added to Rattle: A demo using Kaggle Competition Credit Card Fraud Detection: https://blog.revolutionanalytics.com/2017/07/xgboost-support-added-to-rattle.html (https://blog.revolutionanalytics.com/2017/07/xgboost-support-added-to-rattle.html)

https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/HousePriceAnalysisPrediction/codeR/House%20prices_%20Lasso%2C%20XGBoost%2C% (https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/HousePriceAnalysisPrediction/codeR/House%20prices_%20Lasso%2C%20XGBoost%2C%

https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/Google%20Analytics%20Customer%20Revenue%20Prediction/code/Google%20Analytics% (https://github.com/dd-consulting/DDC-Data-Science-R/blob/master/Google%20Analytics%20Customer%20Revenue%20Prediction/code/Google%20Analytics%

In [88]:
```
# Write your code below and press Shift+Enter to execute
```

Double-click **here** for the solution.

()

Excellent! You have completed the workshop notebook!

Connect with the author:

This notebook was written by GU Zhan (Sam) (https://sg.linkedin.com/in/zhan-gu-27a82823).

Sam (https://www.iss.nus.edu.sg/about-us/staff/detail/201/GU%20Zhan) is currently a lecturer in Institute of Systems Science (https://www.iss.nus.edu.sg/) in National University of Singapore (http://www.nus.edu.sg/). He devotes himself into pedagogy & andragogy, and is very passionate in inspiring next generation of artificial intelligence lovers and leaders.

OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

()

Appendices

Interactive workshops: < Learning R inside R > using swirl() (in R/RStudio)

https://github.com/telescopeuser/S-SB-Workshop (https://github.com/telescopeuser/S-SB-Workshop)

https://github.com/dd-consulting (https://github.com/dd-consulting)