Análise Exploratória

Dsc. Nauber Gois
Fametro
Serviço Federal de Processamento de Dados

Apresentação

Francisco Nauber Bernardo Gois

Analista aprendizado de máquina no Serviço Federal de Processamento de Dados

Doutor em Informática Aplicada

Mestre em Informática Aplicada

Especialista em desenvolvimento WEB

Dúvidas: naubergois@gmail.com

Exploratory Statistics

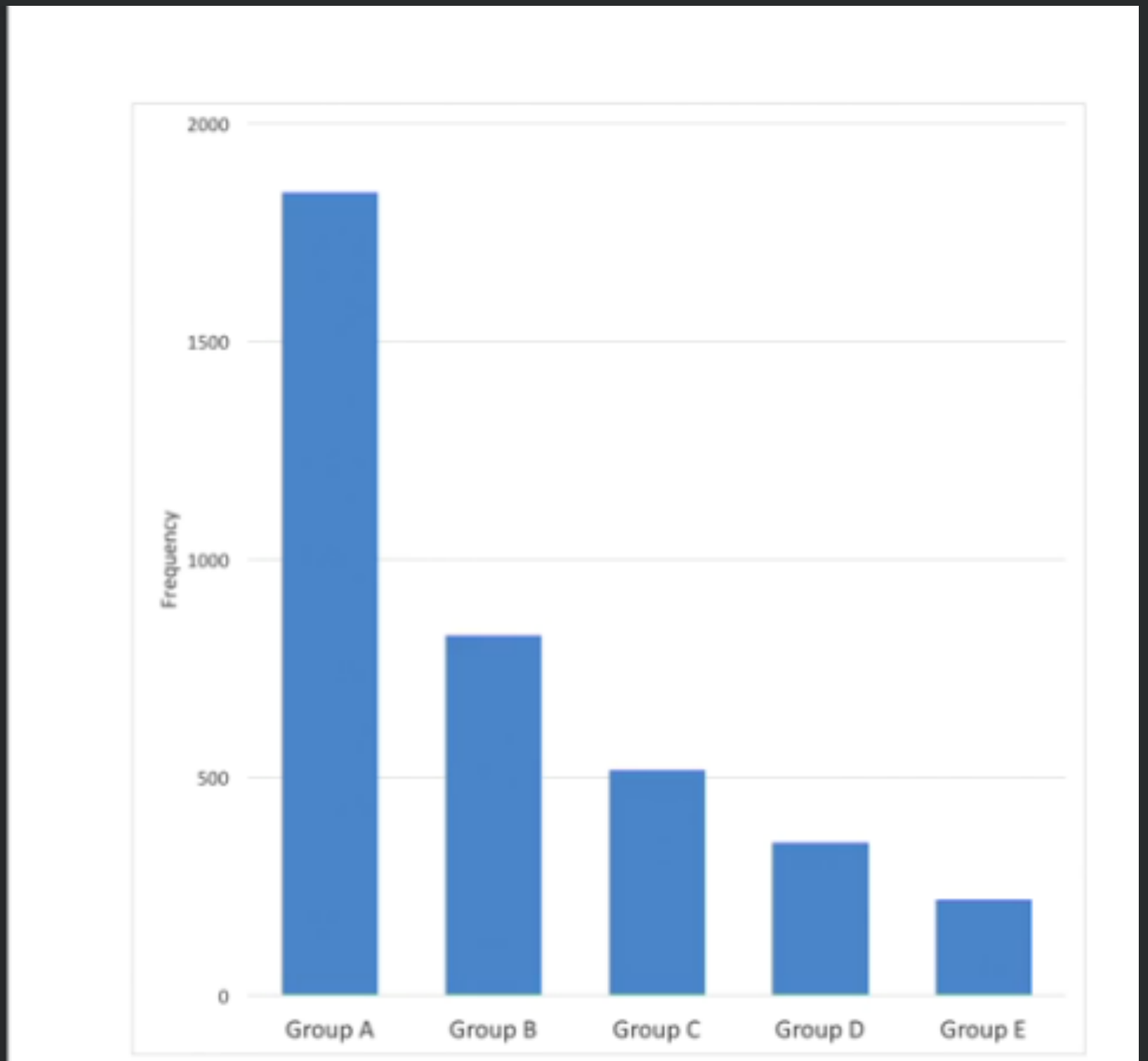
First graphics, then numbers

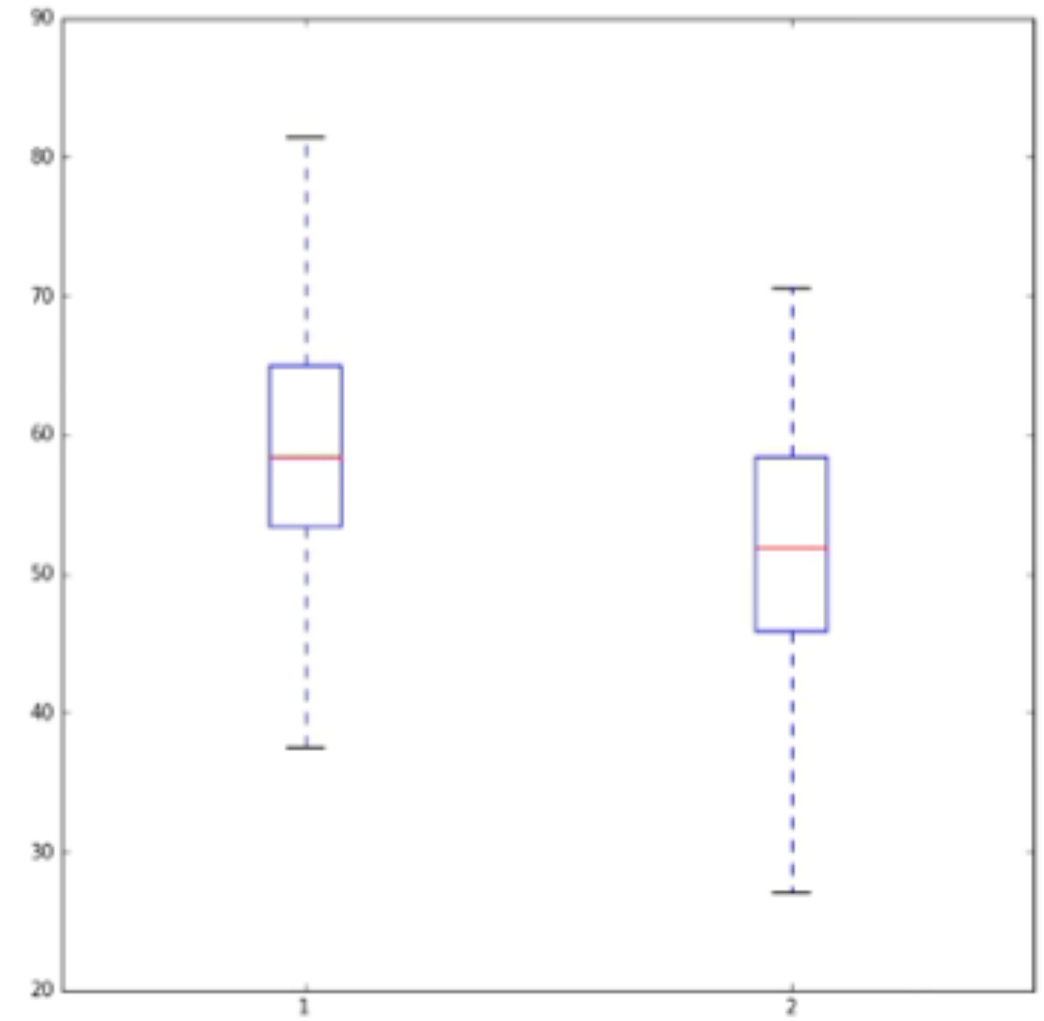
- **Análise de Informações de Forma Visual**
- **Rápida verificação de outiers**

- Single distributions
- Joint distributions
- Unusual cases
- Errors in data
- Missing data

Bar Charts

- For categories
- Easy to read
- Descending values
- Grouped



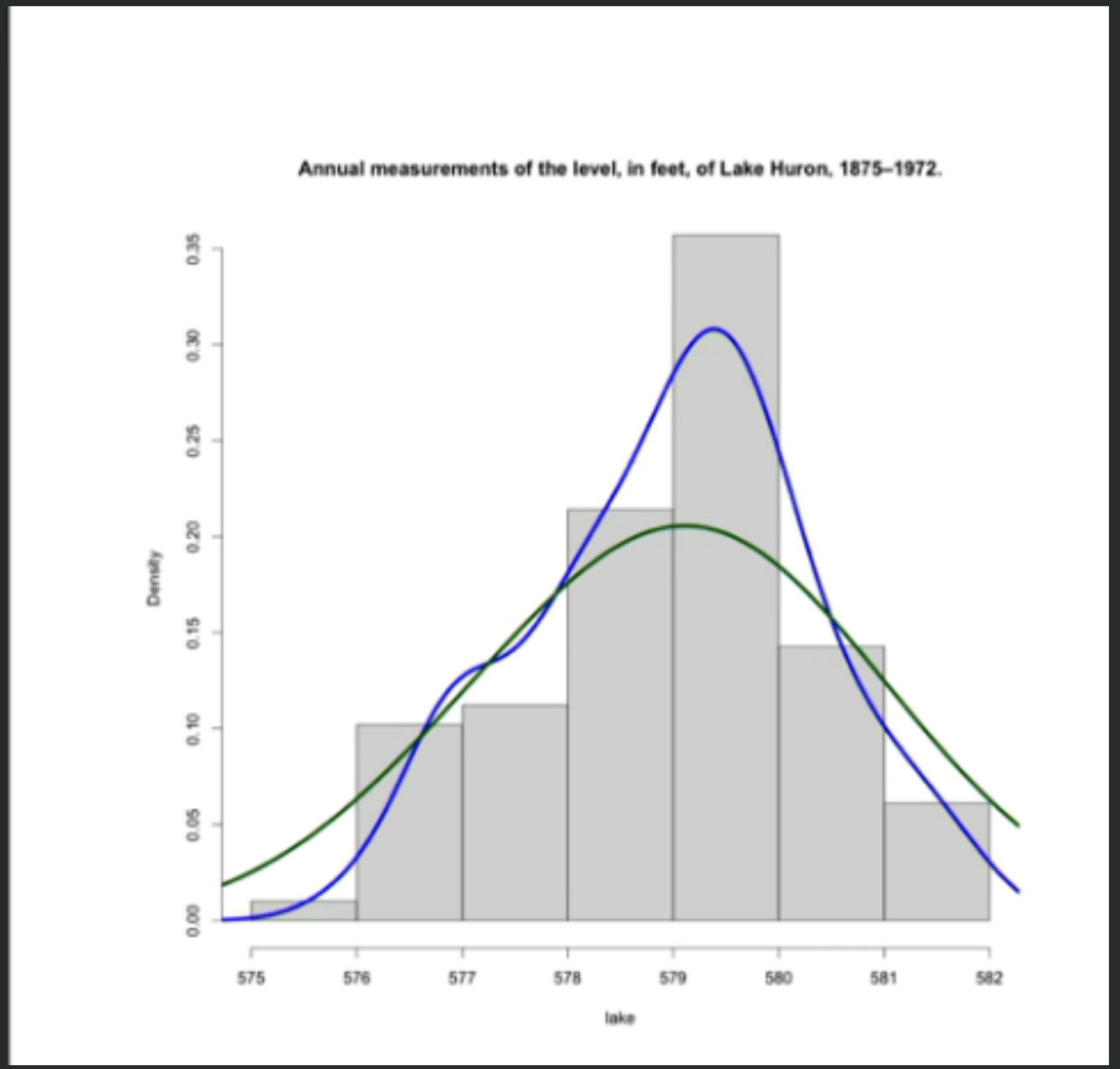


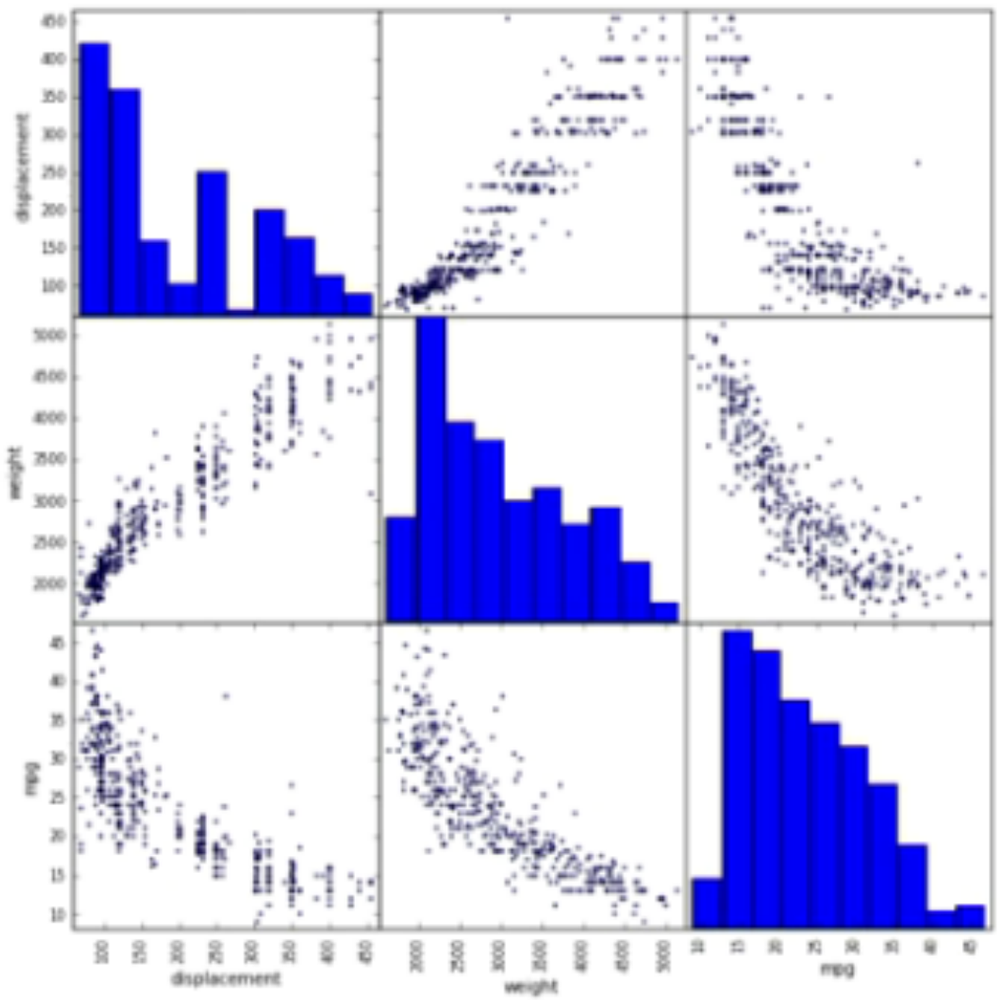
Box Plots

- For quantitative variables
- Shows quartile values and outliers
- Grouped

Histograms

- For quantitative variables
- Shows shape of distribution
- Compare shapes

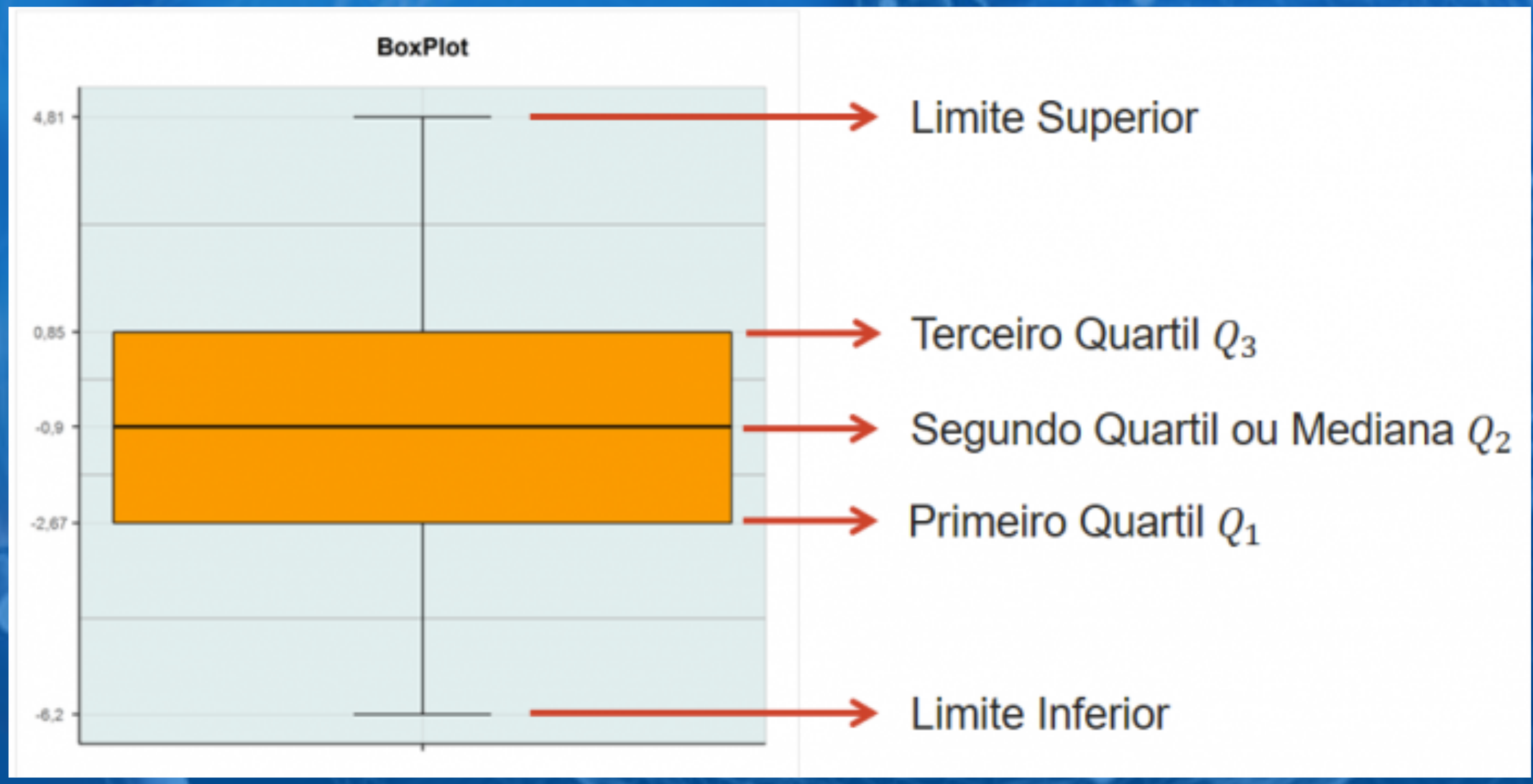




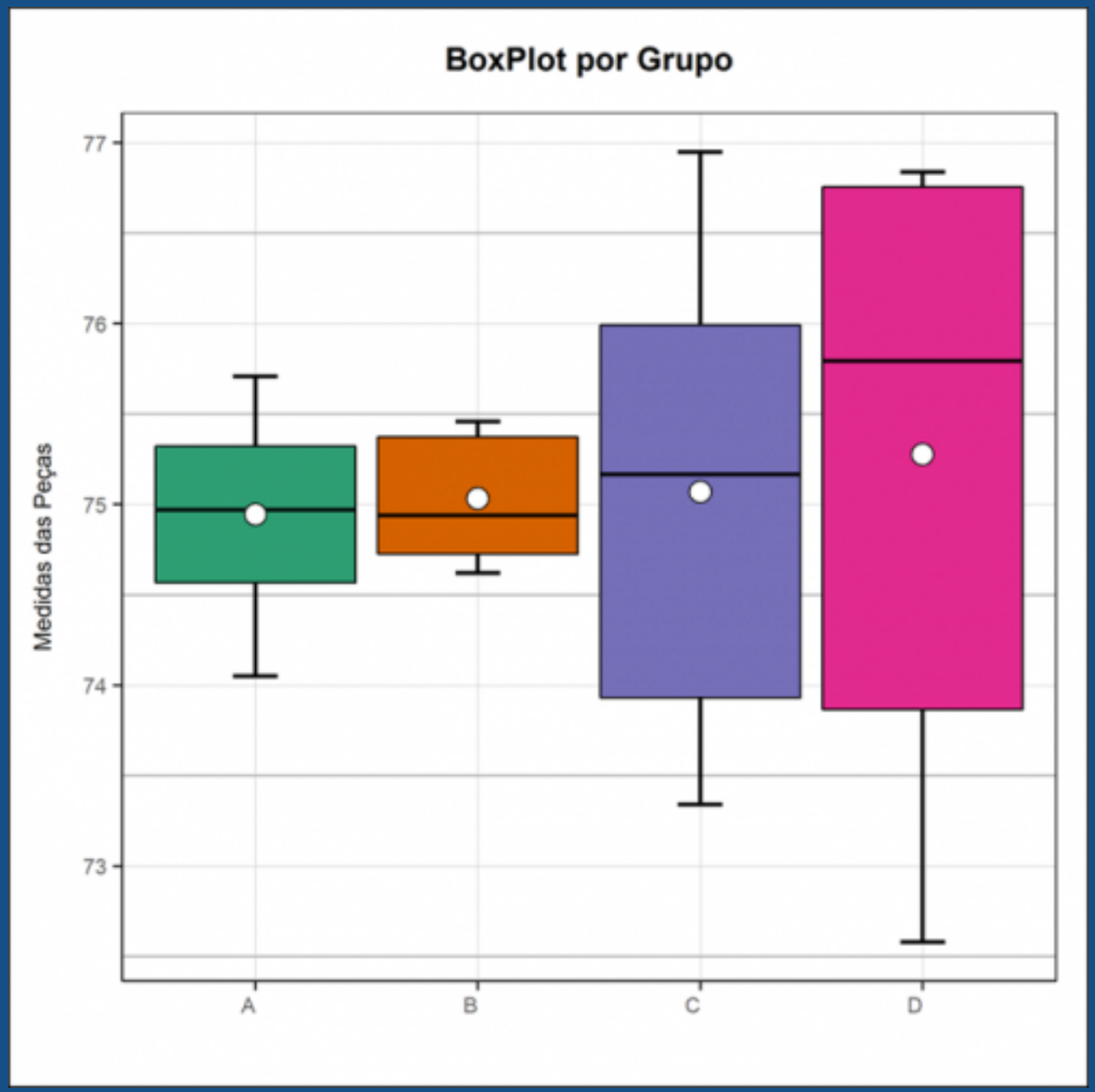
Scatter Plot Matrices

- Show association between quantitative variables
- Easier to read than 3D charts

O boxplot (gráfico de caixa) é um gráfico utilizado para avaliar a distribuição empírica do dados. O boxplot é formado pelo primeiro e terceiro quartil e pela mediana.



```
%matplotlib inline

import numpy as np
import matplotlib as mpl
#mpl.use('agg')
import matplotlib.pyplot as plt

np.random.seed(129)
data_1 = np.random.normal(62, 13, 60)
data_2 = np.random.normal(55, 9, 60)

combined_data = [data_1, data_2]

# figure instance
fig = plt.figure(1, figsize=(9, 9))

# axes instance
ax = fig.add_subplot(111)

# draw boxplot
bxp = ax.boxplot(combined_data)
```



R é uma linguagem e ambiente para computação estatística e gráficos. É um projeto GNU que é semelhante ao idioma e ao ambiente S que foi desenvolvido na Bell Laboratories (anteriormente AT & T, agora Lucent Technologies) por John Chambers e colegas. R pode ser considerado como uma implementação diferente de S. Existem algumas diferenças importantes, mas muito código escrito para S é executado inalterado sob R.

A brief history of R

- **1993**: Research project in Auckland, NZ
  - Ross Ihaka and Robert Gentlemen
- **1995**: Released as open-source software
  - Generally compatible with the "S" language
- **1997**: R core group formed
- **2000**: R 1.0.0 released
- **2004**: First international user conference in Vienna
- **2013**: R 3.0.0 released

```
8 + 5  # Basic math; press cmd/ctrl-enter

1:250  # Prints numbers 1 to 100 across several lines

print("Hello World!")  # Prints "Hello World" in console

# Variables
x <- 1:5  # Put the numbers 1-5 in the variable x
x  # Displays the values in x

y <- c(6, 7, 8, 9, 10)  # Puts the numbers 6-10 in y
y

a <- 1  # Use <- and not =
2 -> a  # Can go other way, but silly
a <- b <- c <- 3  # Multiple assignments

# Vector Math
x
y
x + y  # Adds corresponding elements in x and y
x * 2  # Multiplies each element in x by 2



# Clean up
rm(x)  # Remove an object from workspace
rm(a, b)  # Remove more than one
rm(list = ls())  # Clear entire workspace
```

```
search()   # Shows packages that are currently loaded


install.packages("ggplot2")  # Downloads package from
library("ggplot2")  # Make package available; often used
require("ggplot2")  # Preferred for loading in functions;
library(help = "ggplot2")  # Brings up documentation in
editor window
```

http://www.r-fiddle.org/

```
# Simple Histogram
hist(mtcars$mpg)
```

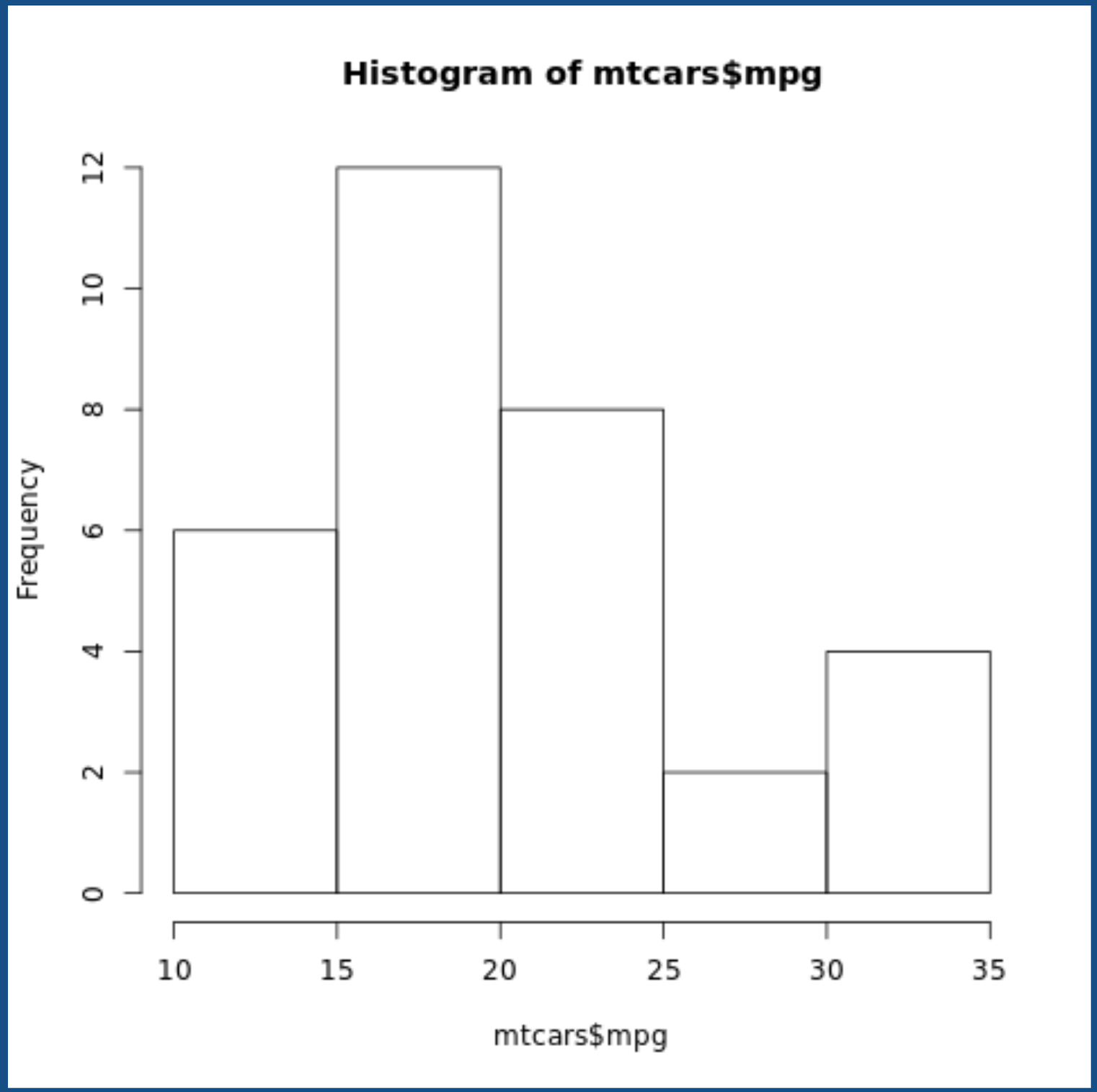


Table vs DataFrame no R

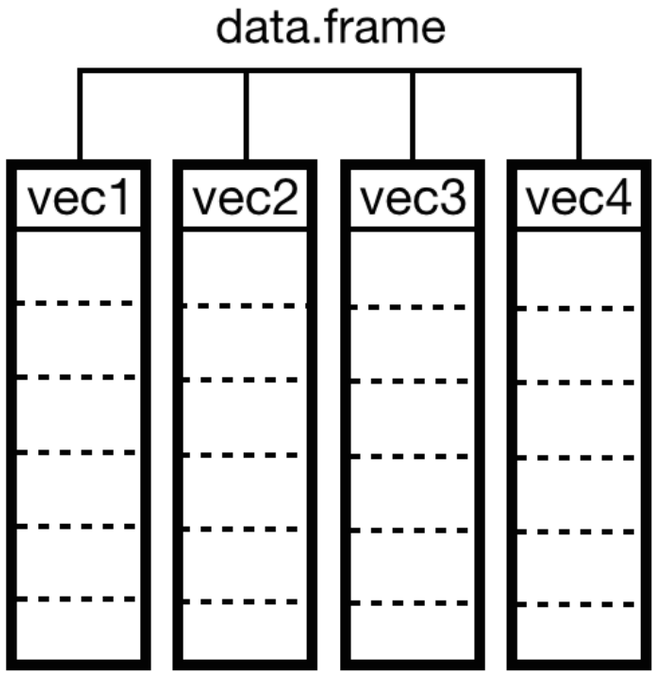


table1

country	year	cases	pop
Afghan	1999	745	19987071
Afghan	2000	2666	20595360
Brazil	1999	37737	172006362
Brazil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

Gráfico em Barra

Exercise Files > Ch02 > 02_01 > Ex02_01.R

R02_01

```
# R Statistics Essential Training
# Ex02_01
# Creating bar charts for categorical variables

# HELP ON PLOTS
?plot

# LOAD DATASETS PACKAGE
require("datasets")

# ONE ROW PER CASE
?chickwts
chickwts  # Look at data
data(chickwts)  # Load data into workspace

# Quickest Method
plot(chickwts$feed)  # Plot feed from chickwts
?plot

# "barplot" offers more control but must prepare data:
# R doesn't create bar charts directly from the categorical
# variables; instead, we must first create a table that
# has the frequencies for each level of the variable.
feeds <- table(chickwts$feed)
feeds
barplot(feeds)  # Identical to plot(chickwts$feed)
?barplot

# To put the bars in descending order, add "order":
barplot(feeds[order(feeds, decreasing = TRUE)])

# Customize the chart
par(oma = c(1, 1, 1, 1))  # Sets outside margins: b, l, t, r
par(mar = c(4, 5, 2, 1))  # Sets plot margins
barplot(feeds[order(feeds)],
        horiz  = TRUE,
        las    = 1,  # las gives orientation of axis labels
        col    = c("beige", "blanchedalmond", "bisque1", "bisque2", "bisque3", "bisque4"),
        border = NA,  # No borders on bars
        main   = "Frequencies of Different Feeds\nin chickwts Dataset",  # \n = line break
        xlab   = "Number of Chicks")
?par

rm(list = ls())  # Clean up
```

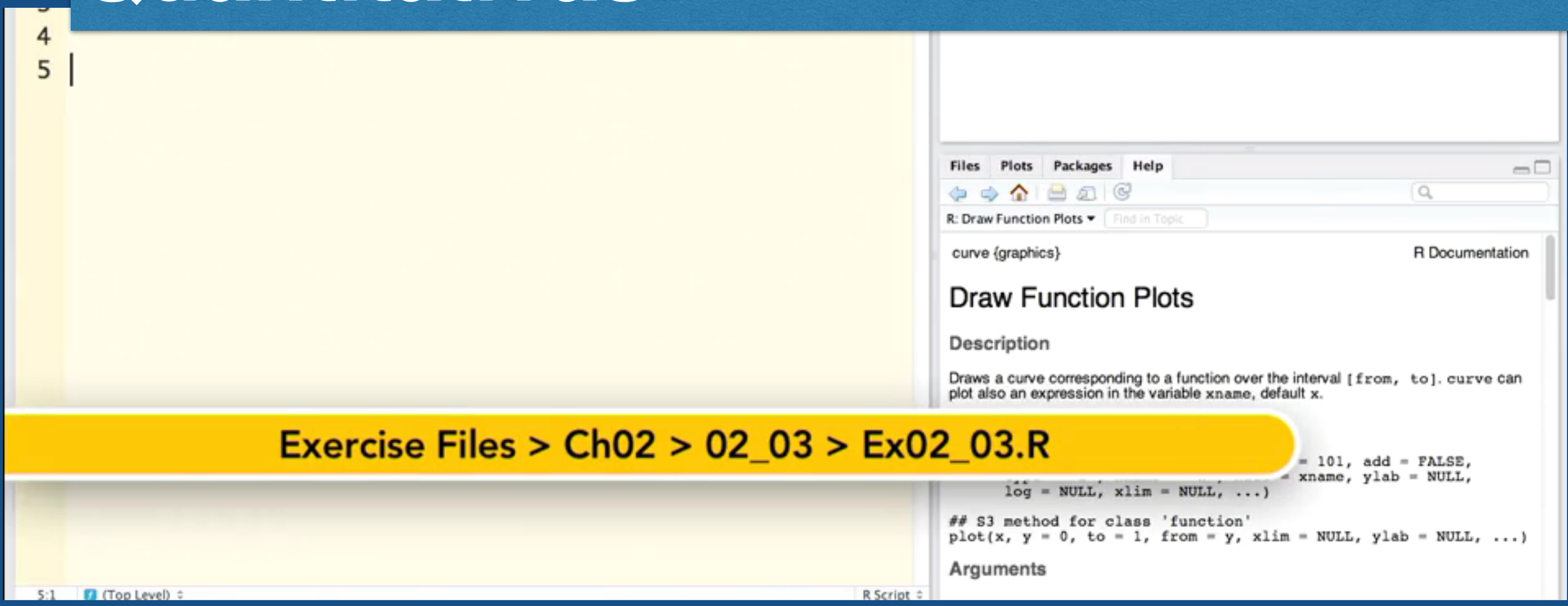
Criando Diagrama de Pizza

Exercise Files > Ch02 > 02_02 > Ex02_02.R

R02_02

```
# R Statistics Essential Training
# Ex02_02
# Creating pie charts for categorical variables

# LOAD DATASETS PACKAGE
require("datasets")

# ONE ROW PER CASE
data(chickwts)

# Create a table with frequencies
feeds <- table(chickwts$feed)
feeds

# Make the pie chart with the defaults
pie(feeds)
?pie

# Modify the pie chart
pie(feeds[order(feeds, decreasing = TRUE)],
    init.angle = 90,   # Starts as 12 o'clock instead of 3
    clockwise = TRUE,  # Default is FALSE
    col = c("seashell", "cadetblue2", "lightpink", "lightcyan", "plum1", "papayawhip"),
    main = "Pie Chart of Feeds from chickwts")

# THE PROBLEM WITH PIE CHARTS
# Three data sets
pie.a <- c(22, 14, 18, 20, 14, 12)
pie.b <- c(20, 18, 16, 18, 16, 12)
pie.c <- c(12, 14, 20, 18, 14, 22)

# Changing graphical parameters for a minute
oldpar <- par()   # Stores old graphical parameters
par(mfrow    = c(1, 3),  # Num. rows/cols
    cex.main = 3)   #  Main title 3x bigger
colors <- c("grey98", "grey90", "lightskyblue", "lightgreen", "grey98", "grey90")
?colors

# Three pie charts side by side
# Is the green slice or blue slice bigger?
pie(pie.a, main = "Pie A", col = colors)
pie(pie.b, main = "Pie B", col = colors)
pie(pie.c, main = "Pie C", col = colors)

# Three bar charts side by side
# Is the green bar or blue bar bigger?
barplot(pie.a, main = "Bar A", col = colors)
barplot(pie.b, main = "Bar B", col = colors)
barplot(pie.c, main = "Bar C", col = colors)

# CONCLUSION
# From R help on pie charts:
?pie

# Pie charts are a very bad way of displaying information.
# The eye is good at judging linear measures and bad at
# judging relative areas. A bar chart or dot chart is a
# preferable way of displaying this type of data.
#
# Cleveland (1985), page 264: "Data that can be shown by
# pie charts always can be shown by a dot chart. This means
# that judgements of position along a common scale can be
# made instead of the less accurate angle judgements."
# This statement is based on the empirical investigations
# of Cleveland and McGill as well as investigations by
# perceptual psychologists.

par(oldpar)  # Restore old graphical parameters
# Note that cin, cra, csi, cxy, and din are read-only
# parameters that were written to oldpar but cannot be
# rewritten; just ignore the warning messages for these.
?par

rm(list = lm())  # Clean up
```

Histograma para variaveis Quantitativas



Exercise Files > Ch02 > 02_03 > Ex02_03.R

R02_03

```
require("datasets")
?lynx
data(lynx)  # Annual Canadian Lynx trappings 1821-1934

# Make a histogram using the defaults
hist(lynx)
?hist

# Modify histogram
h <- hist(lynx,  # Save histogram as object
        breaks = 11,  # "Suggests" 11 bins
#         breaks = seq(0, 7000, by = 100),
#         breaks = c(0, 100, 300, 500, 3000, 3500, 7000),
        freq = FALSE,
        col = "thistle1", # Or use: col = colors() [626]
        main = "Histogram of Annual Canadian Lynx Trappings\n1821-1934",
        xlab = "Number of Lynx Trapped")

# IF freq = FALSE, this will draw normal distribution
curve(dnorm(x, mean = mean(lynx), sd = sd(lynx)),
      col = "thistle4",
      lwd = 2,
      add = TRUE)
?curve

rm(list = ls())  # Clean up
```

Box Plot para variáveis Quantitativas

Files Plots Packages Help

Zoom Export Clear All

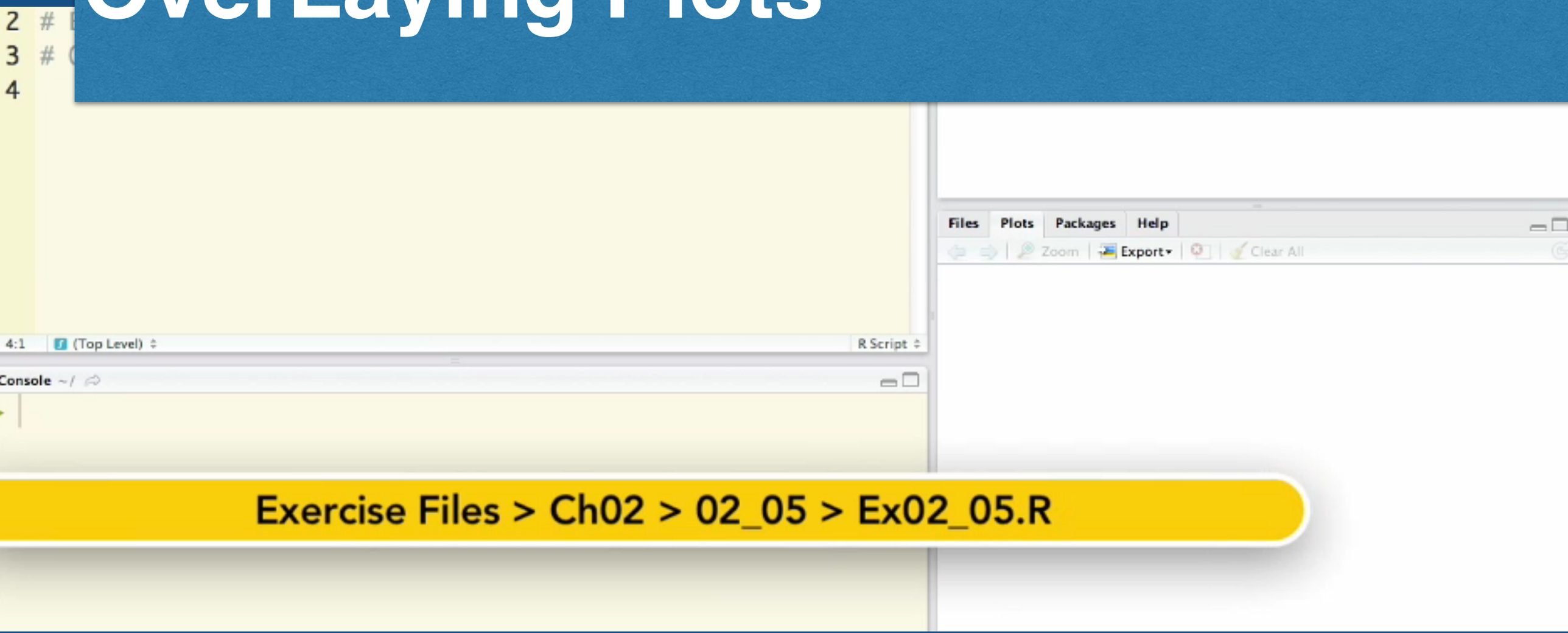
Exercise Files > Ch02 > 02_04 > Ex02_04.R

5:1 (Top Level) R Script

```
# R Statistics Essential Training
# Ex02_04
# Creating boxplots for quantitative variables

# LOAD DATASET
require("datasets")
# Lawyers' Ratings of State Judges in the US Superior Court (c. 1977)
?USJudgeRatings
USJudgeRatings  # View data
data(USJudgeRatings)  # Load into workspace
# At least two errors in data file:
# 1. Data appears to be on 1-10 or 0-10 scale but Callahan
#    has a 10.6 on CONT. 8.6 seems more likely.
# 2. Santaniello's last name is misspelled
# Best to fix errors in spreadsheet and reimport

# Make boxplot using the defaults
boxplot(USJudgeRatings$RTEN)
?boxplot

# Modify boxplot
boxplot(USJudgeRatings$RTEN,
        horizontal = TRUE,
        las = 1,  # Make all labels horizontal
        notch = TRUE,  # Notches for CI for median
        ylim = c(0, 10),  # Specify range on Y axis
        col = "slategray3",   # R's named colors (n = 657)
#       col = colors() [602], # R's color numbers
#       col = "#9FB6CD",      # Hex codes for RBG
#       col = rgb(159, 182, 205, max = 255),  # RGB triplet with max specified
        boxwex = 0.5,  # Width of box as proportion of original
        whisklty = 1,  # Whisker line type; 1 = solid line
        staplelty = 0,  # Staple (line at end) type; 0 = none
        outpch = 16,  # Symbols for outliers; 16 = filled circle
        outcol = "slategray3",  # Color for outliers
        main = "Lawyers' Ratings of State Judges in the\nUS Superior Court (c. 1977)",
        xlab = "Lawyers' Ratings")

# Multiple boxplots
boxplot(USJudgeRatings,
        horizontal = TRUE,
        las = 1,  # Make all labels horizontal
        notch = TRUE,  # Notches for CI for median
        ylim = c(0, 10),  # Specify range on Y axis
        col = "slategray3",   # R's named colors (n = 657)
        boxwex = 0.5,  # Width of box as proportion of original
        whisklty = 1,  # Whisker line type; 1 = solid line
        staplelty = 0,  # Staple (line at end) type; 0 = none
        outpch = 16,  # Symbols for outliers; 16 = filled circle
        outcol = "slategray3",  # Color for outliers
        main = "Lawyers' Ratings of State Judges in the\nUS Superior Court (c. 1977)",
        xlab = "Lawyers' Ratings")

rm(list = ls())  # Clean up
```

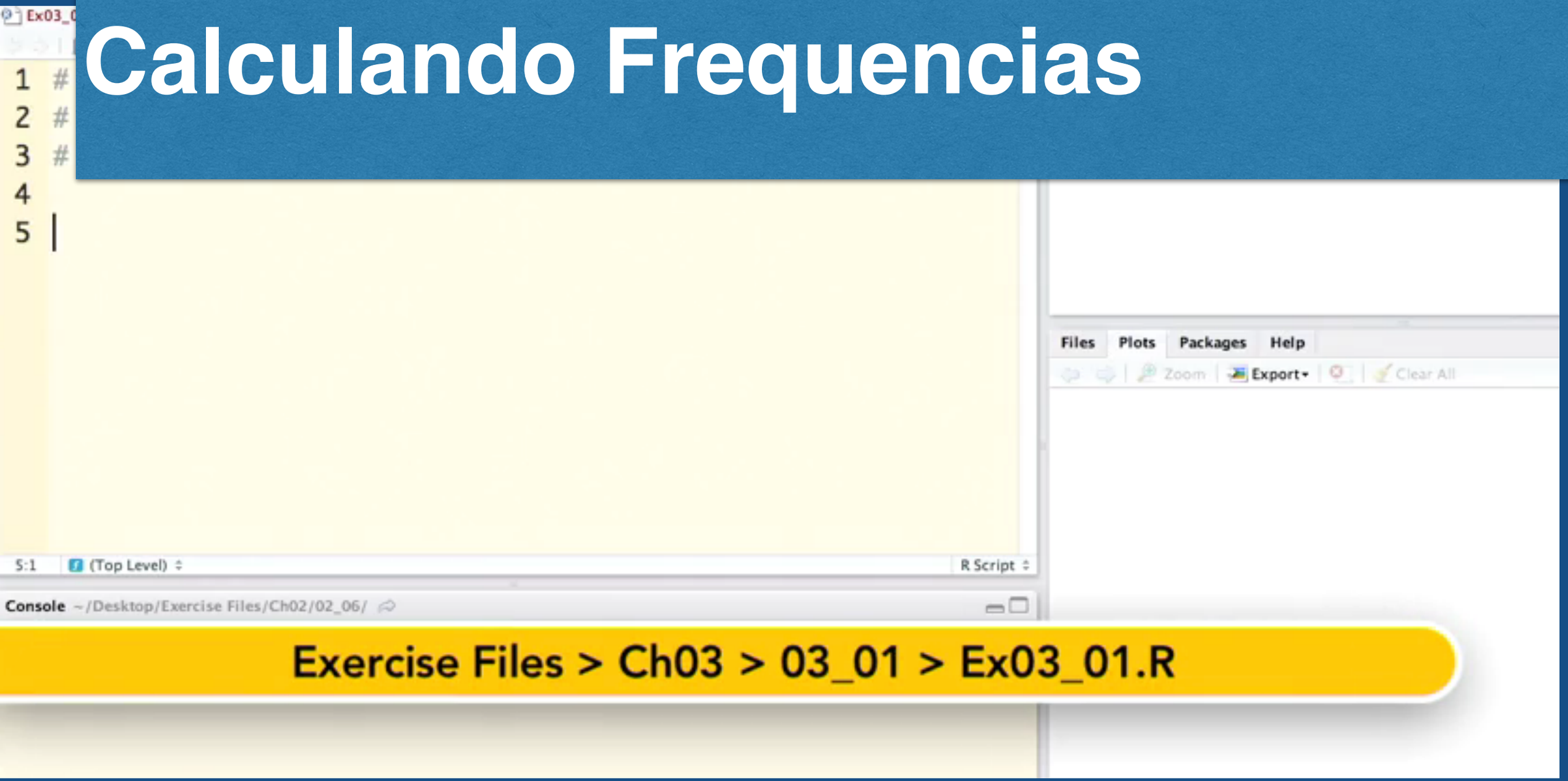
R02_04

OverLaying Plots



Exercise Files > Ch02 > 02_05 > Ex02_05.R

Calculando Frequencias

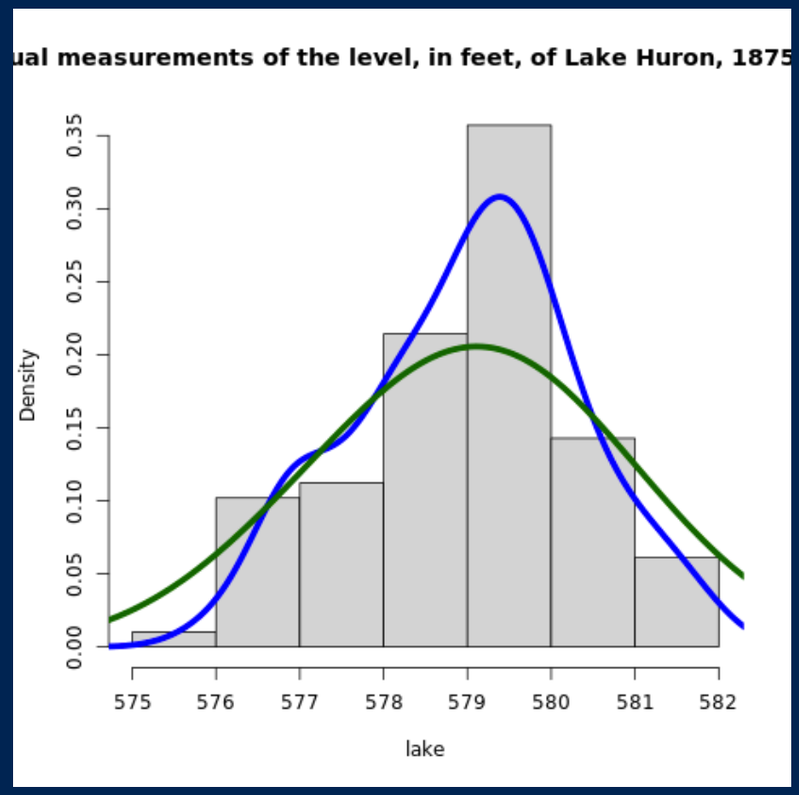


Exercise Files > Ch03 > 03_01 > Ex03_01.R

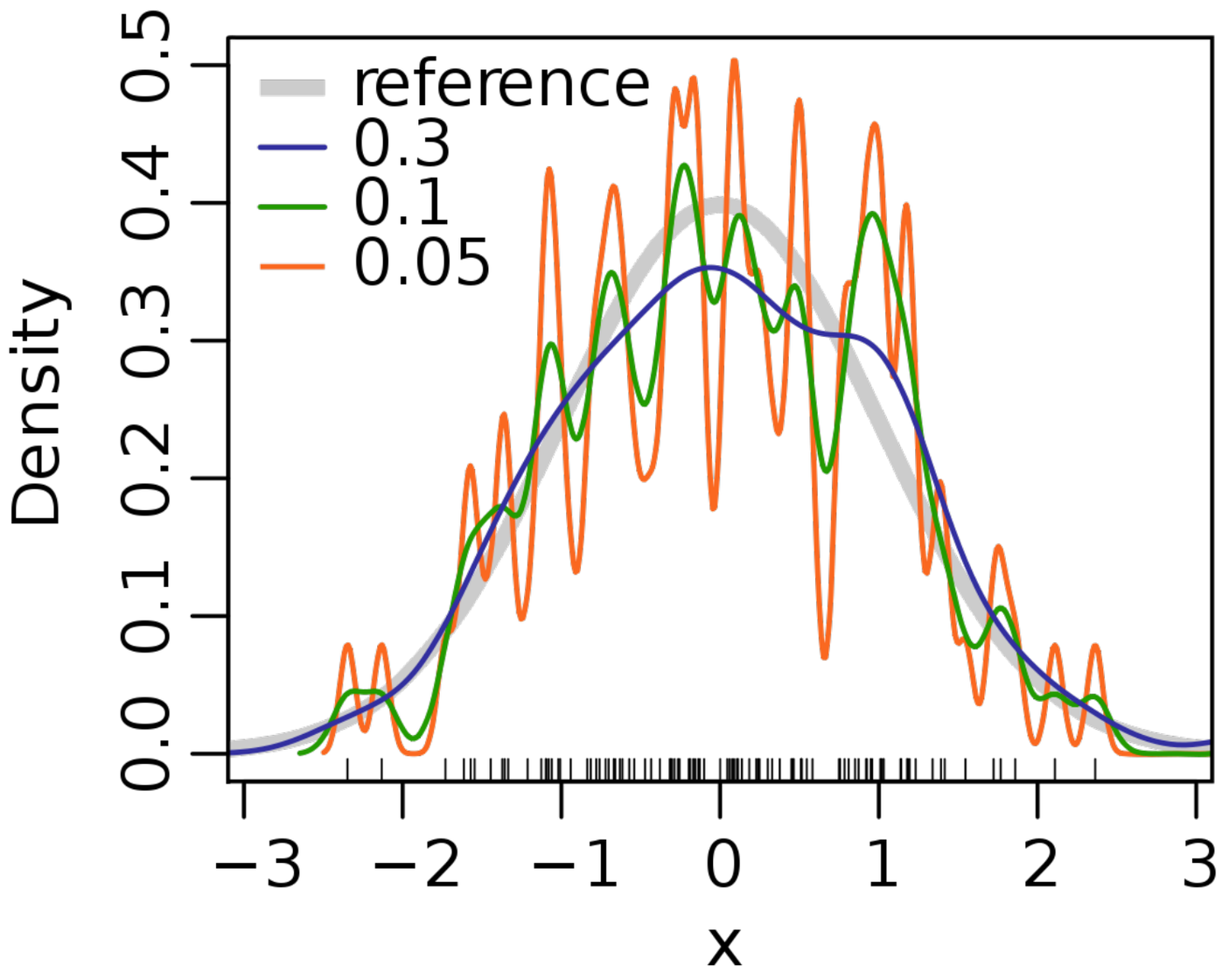
```
groups <- c(rep("blue",   3990),
            rep("red",    4140),
            rep("orange", 1890),
            rep("green",  3770),
            rep("purple",  855))

# CREATE FREQUENCY TABLES
groups.t1 <- table(groups)  # Creates frequency table
groups.t1  # Print table

# MODIFY FREQUENCY TABLES
groups.t2 <- sort(groups.t1, decreasing = TRUE)  # Sorts by frequency, saves table
groups.t2  # Print table

# PROPORTIONS AND PERCENTAGES
prop.table(groups.t2)  # Give proportions of total
round(prop.table(groups.t2), 2)  # Give proportions w/2 decimal places
round(prop.table(groups.t2), 2) * 100  # Give percentages w/o decimal places

rm(list = ls())  # Clean up
```

http://www.r-fiddle.org/

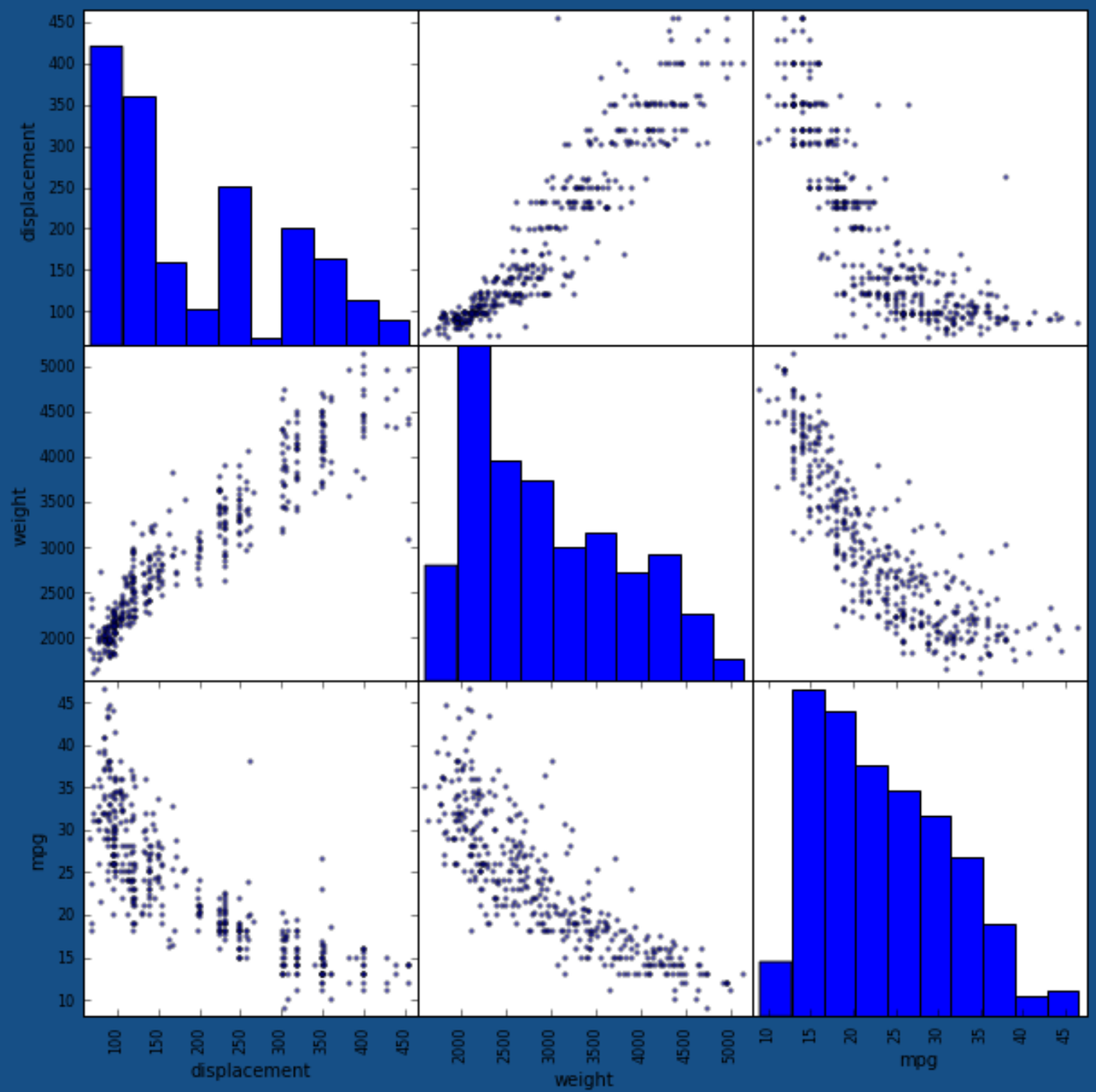
```
require("datasets")

# Histogram
lake <- LakeHuron
hist(lake)

# LOAD DATASET
require("datasets")
data(lake)  # Annual Canadian Lynx trappings 1821-1934

# HISTOGRAM WITH OPTIONS
hist(lake,
     freq = FALSE,  # Axis shows density, not frequency
     col = "light gray",  # Color for histogram
     main = "Annual measurements of the level, in feet, of Lake Huron, 1875–1972.")

# SUPERIMPOSED KERNEL DENSITY ESTIMATES
lines(density(lake), col = "blue", lwd = 5)
lines(density(lake, adjust = 3), col = "darkgreen", lwd = 5)
```

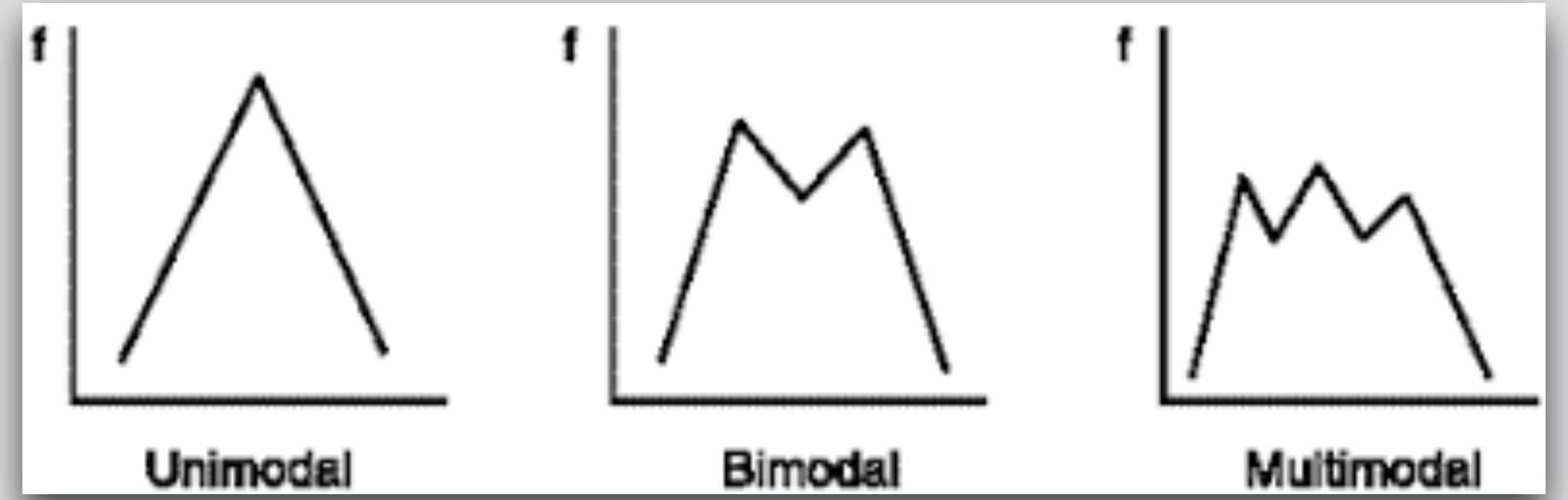


In statistics, kernel density estimation (KDE) is a non-parametric way to estimate the probability density function of a random variable.



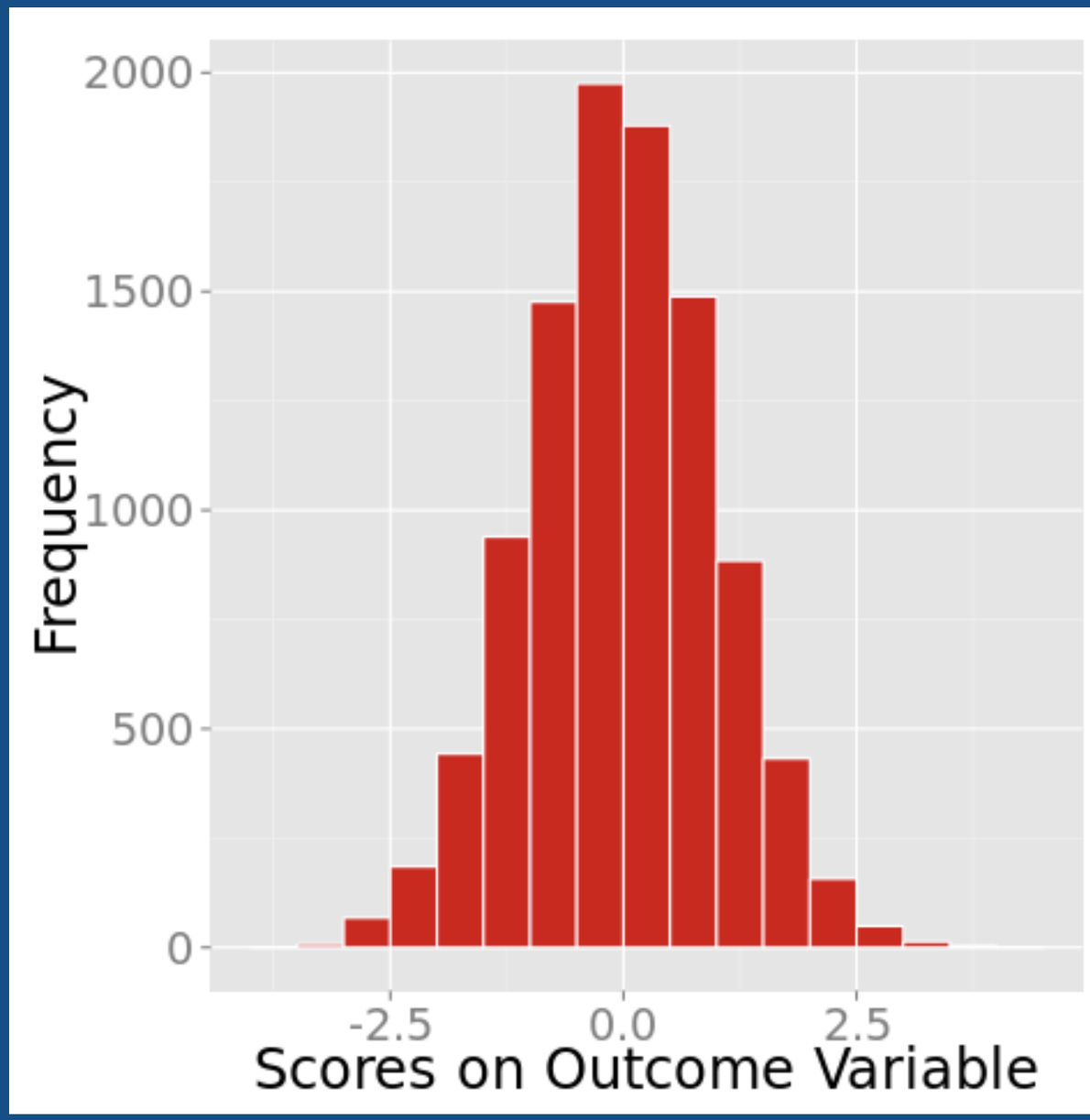
```
import pandas as pd
from pandas.tools.plotting import scatter_matrix as spm

cars = pd.read_csv("http://www-bcf.usc.edu/~gareth/ISL/Auto.csv")

_ = spm(cars[['displacement', 'weight', 'mpg']], figsize=(10, 10))
```



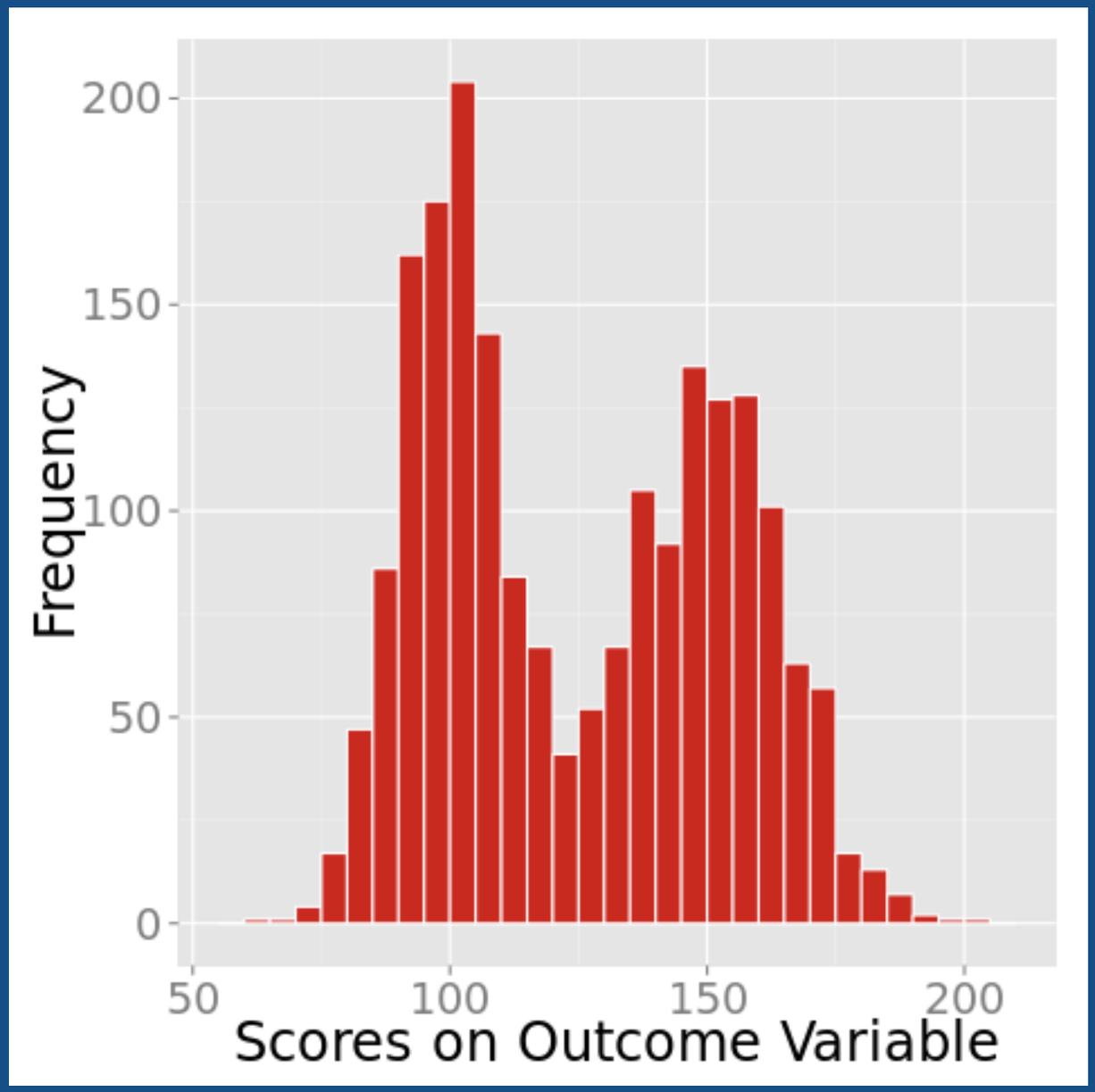


Bimodal Distributions

```
library("ggplot2")
xnorm <- rnorm(10000)
ggplot(NULL, aes(x =
xnorm)) +
  geom_histogram(binwidth
= 0.5,
                 fill = "firebrick3",
                 colour = "white") +
  xlab("Scores on Outcome
Variable") +  # Title on Y axis
  ylab("Frequency") +
  theme(text =
element_text(size = 24))
```



```
library("reshape2")
# - Individual distributions
nd1 <- rnorm(1000, mean = 100,
sd = 10)
nd2 <- rnorm(1000, mean = 150,
sd = 15)
df12   <- data.frame(nd1, nd2)
melt12 <- melt(df12)
hist(melt12$value)
ggplot(melt12, aes(x = value)) +
  geom_histogram(binwidth = 5,
fill = "firebrick3", colour =
"white") +
  xlab("Scores on Outcome
Variable") +  # Title on Y axis
  ylab("Frequency") +
  theme(text = element_text(size
= 24))
```



Hypothesis Testing

Test your theory

Uses of Hypothesis Testing

- Scientific research
- Diagnostics
- Go/no go decisions

TESTE DE HIPÓTES

Trata-se de uma técnica para se fazer a inferência estatística sobre uma população a partir de uma amostra

What is Hypothesis Testing?

Hypothesis testing refers to

1. Making an assumption, called hypothesis, about a population parameter.
2. Collecting sample data.
3. Calculating a sample statistic.
4. Using the sample statistic to evaluate the hypothesis (how likely is it that our hypothesized parameter is correct. To test the validity of our assumption we determine the difference between the hypothesized parameter value and the sample value.)

1

Null Hypothesis

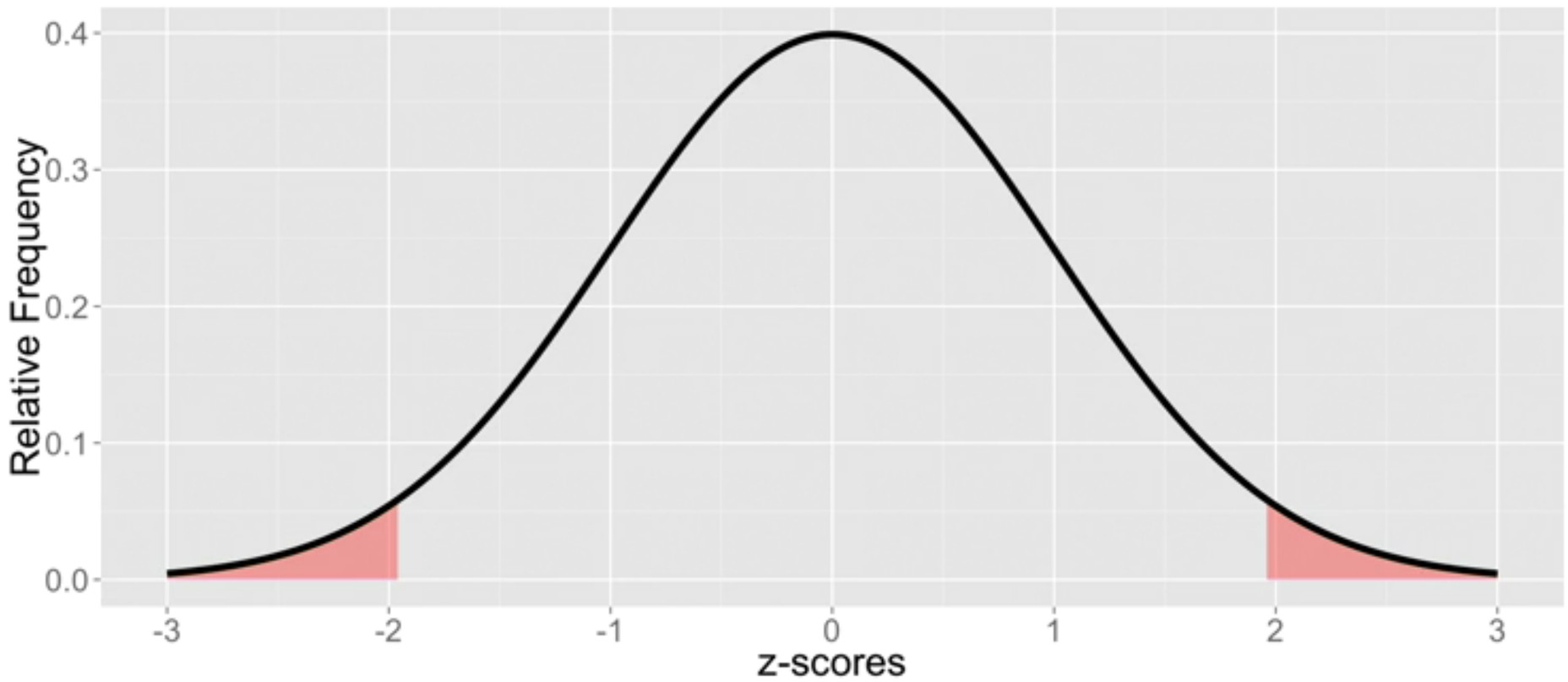
H_0: No systematic effect and random sampling error is only explanation

2

Alternative Hypothesis

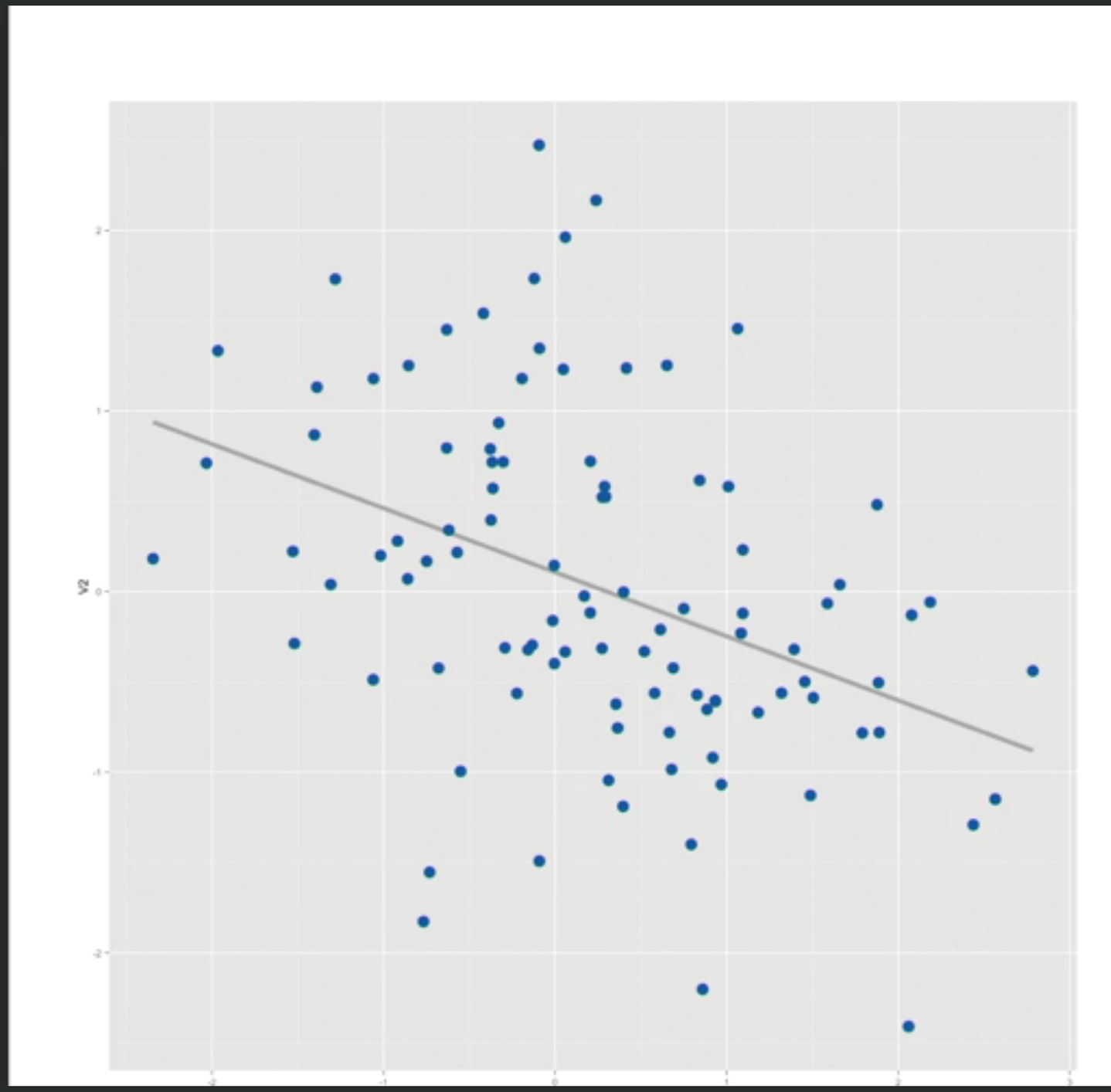
H_A: Systematic effect

Set Regions of Rejection



False Positive

- Sample shows effect, but it's randomness.
- Conditional on rejecting null
- Type I error
- Pick a value: .05



1

Accuracy

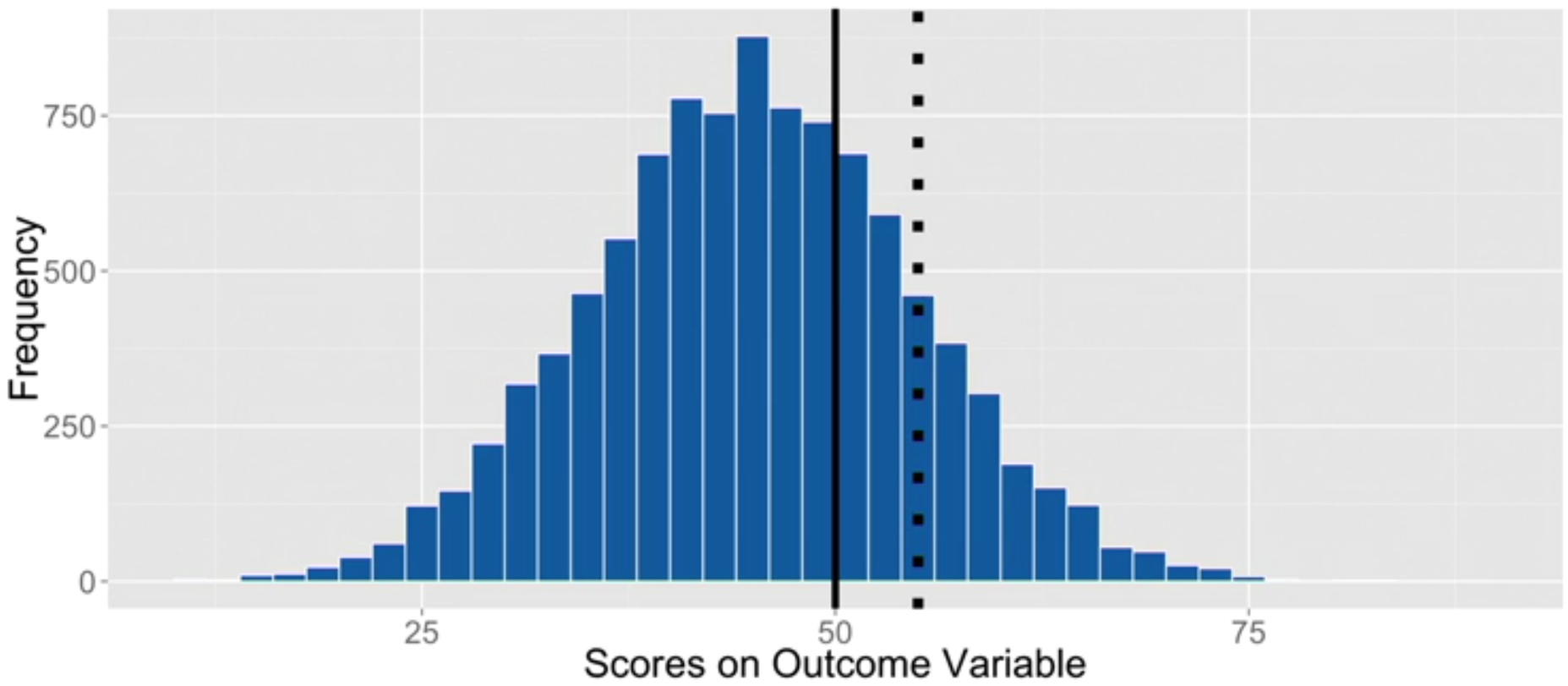
- Accuracy = on target
- Contains the true population value
- Leads to correct inference

2

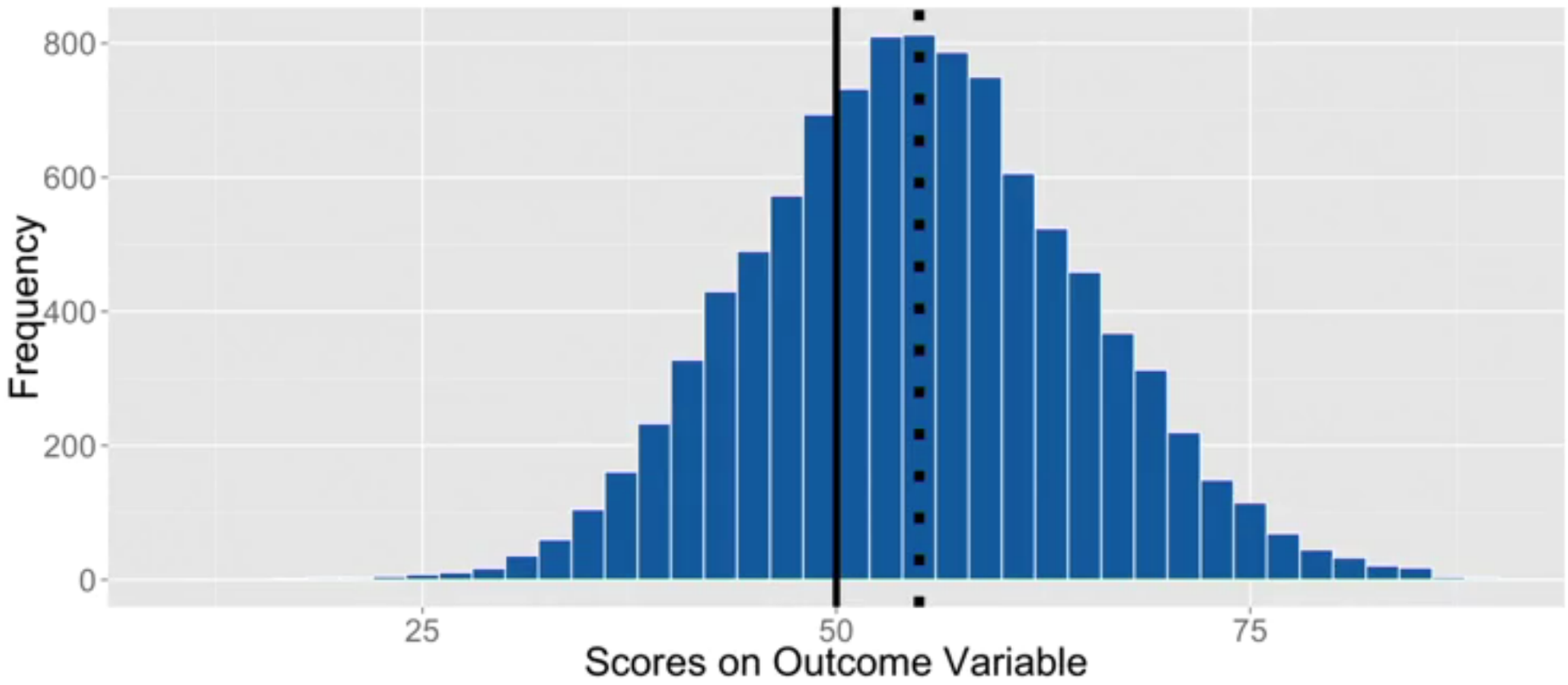
Precision

- Precise = narrow
- Small range of likely values
- Independent of accuracy

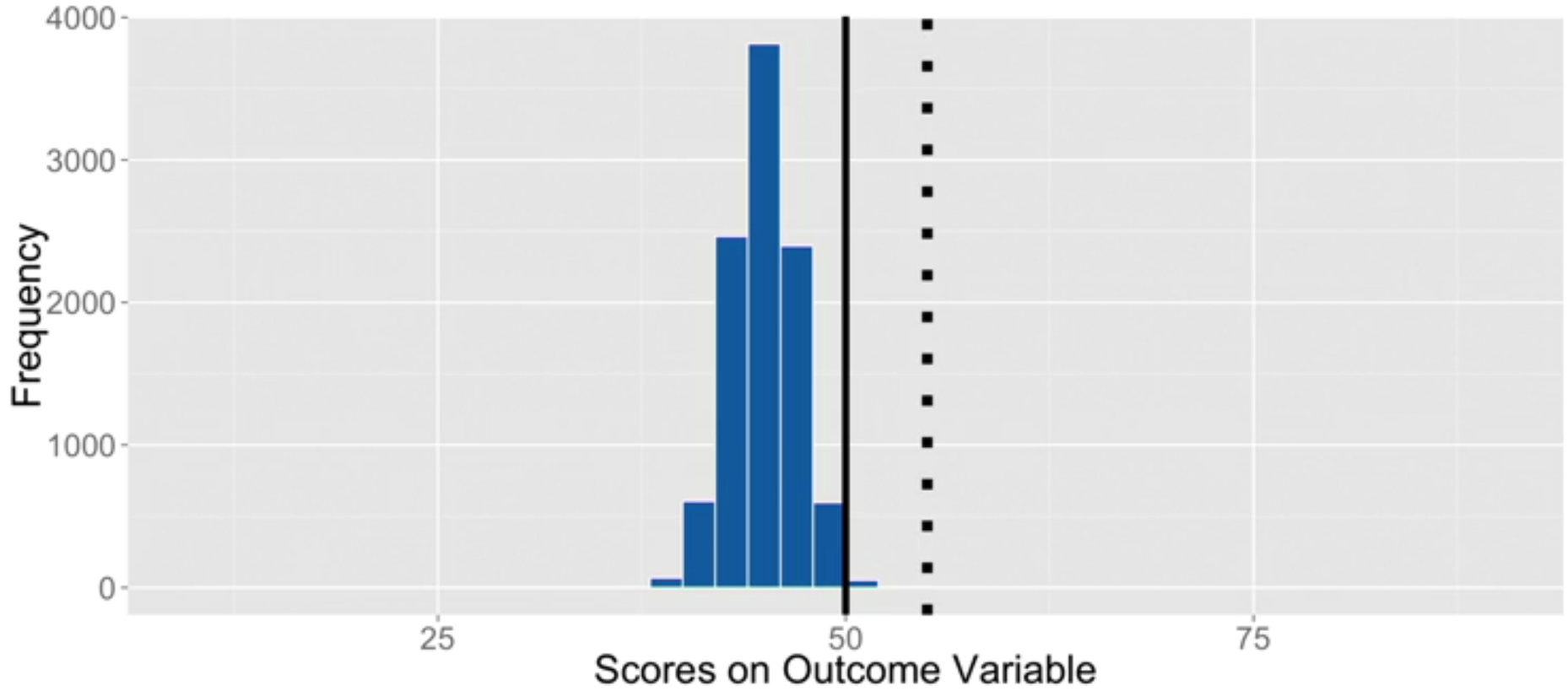
Neither Accurate nor Precise



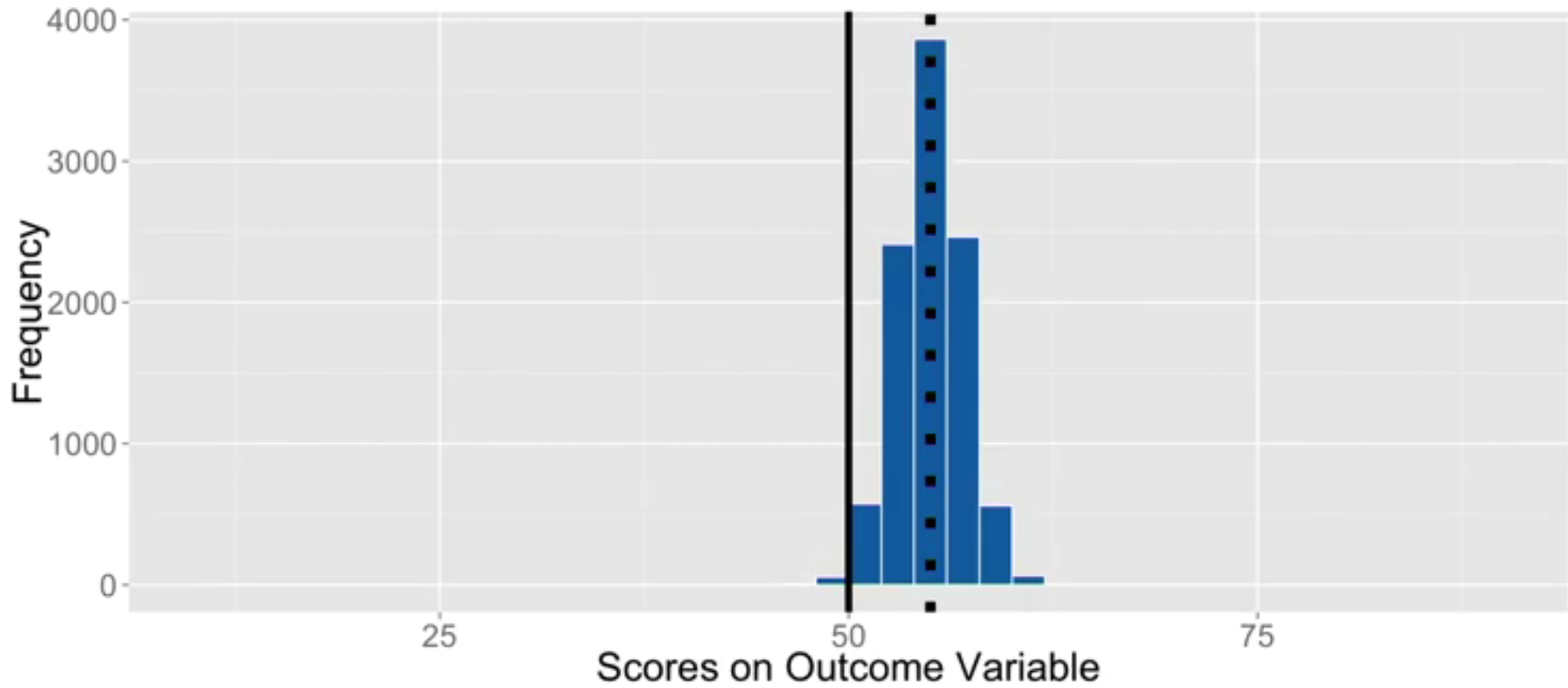
Accurate but Not Precise



Precise but Not Accurate



Accurate and Precise



DISTRIBUIÇÃO NORMAL

```
px <- 1.4   # X coordinate for point on graph
lb <- -1.4  # Lower bounds for marked region
ub <- 1.4   # Upper bounds for marked region
mean = 0    # Mean of distribution
sd = 1      # SD of distribution
limits = c(mean - 3 * sd, mean + 3 * sd)

x <- seq(limits[1], limits[2], length.out = 300)
xmin <- max(lb, limits[1])
xmax <- min(ub, limits[2])
areax <- seq(xmin, xmax, length.out = 300)
area <- data.frame(x = areax,
                   ymin = 0,
                   ymax = dnorm(areax, mean = mean, sd = sd))
ggplot() +
  geom_line(data.frame(x = x, y = dnorm(x, mean = mean, sd = sd)),
            mapping = aes(x = x, y = y), size = 2) +
  scale_x_continuous(limits = limits,
                     breaks = seq(limits[1], limits[2], 1)) +
  xlab("z-scores") +  # Title on Y axis
  ylab("Relative Frequency") +
  theme(text = element_text(size = 24)) +
  theme(legend.position = "none")
```

Decision Trees

Trees are the answer...

1

Classification Trees

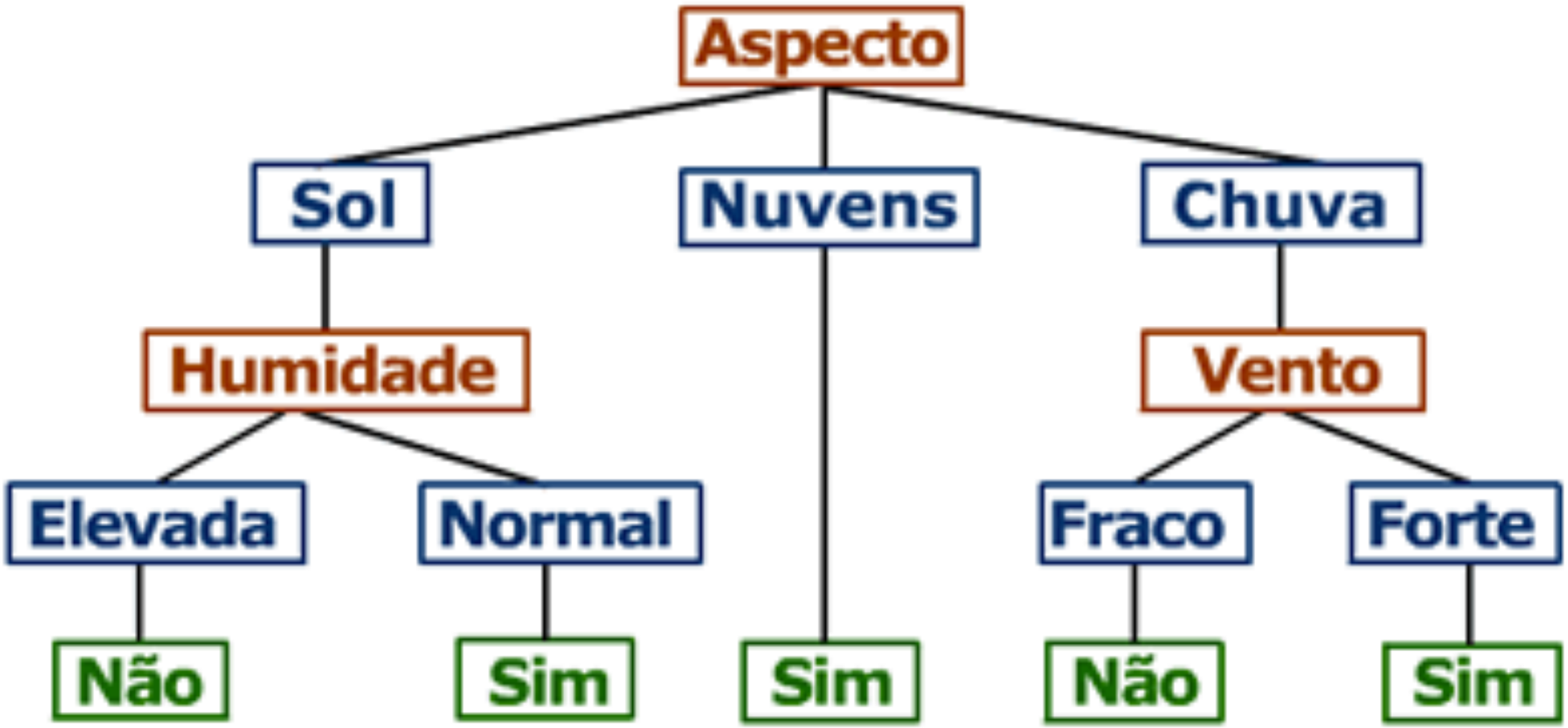
Use quantitative and categorical data to model categorical outcomes

2

Regression Trees

Use quantitative and categorical data to model quantitative outcomes

Árvore de Decisão para Jogar Ténis



```
# IDS_09_01_DecisionTrees.R

# ============================================================
# Install and load packages as needed
# ============================================================

install.packages("party")  # Install party
library(datasets)          # Load built-in datasets
library(party)             # Load party

# Using ctree (conditional inference trees) from party

# ============================================================
# Classification tree
# ============================================================

# Use "iris" from built-in datasets

head(iris)  # Show first six cases

# Create a tree to predict species by measurements

iris.ct <- ctree(Species ~ ., data = iris)  # Create tree
iris.ct                                     # Tree info
plot(iris.ct)                               # Tree plot
table(predict(iris.ct), iris$Species)       # Pred vs true

# ============================================================
# Clean up
# ============================================================

# Clear workspace
rm(list = ls())

# Unload packages
detach("package:datasets", unload = TRUE)
detach("package:party", unload = TRUE)

# Clear plots
dev.off()

# Clear console
cat("\014")  # ctrl+L
```