```r
# Lecture 5a: Computer Problem Solving with R commands

# If you wish to use any of the libraries noted below, then you will need to copy and paste the following commands in R first
install.packages("install.load") # install the install.load package
library(install.load) # load the install.load package (library)
install_load("pracma") # install and load the named packages and all of their dependencies (this process may take a while depending on the number of dependencies)


# The following code is also online at http://www.r-fiddle.org/#/fiddle?id=uHMLVe5b

help(plot)

?plot

help.start()

apropos("plot")

X <- 5

x

X

print(X)

x <- c(2:4)

x

5^2 + 4^2

(5 + 4)^2

(2+3)/(4-5)

log10(100)

log(4*(2+3))

cos(30*pi/180)

sin(30*pi/180)

str(x)

y <- x^3 + 2*x^2 + 0.5*x + 54

str(y)

x <- c(2:4)

y <- x^3 + 2*x^2 + 0.5*x + 54

plot(x, y, main = "Plot of x versus y", xlab = "x", ylab = "y")

x <- c(-1.6, -1.5, -1.4, 1.4, 1.5, 1.6)

min(x)

max(x)

# Example 2. Compute the average temperature and plot the temperature data. (Etter 16-17)
#
time <- c(0.0, 0.5, 1.0)
temps <- c(105, 126, 119)
```

```r
average <- mean(temps)
plot(time, temps, type = "l", main = "Temperature Measurement", xlab = "Time (min)", ylab = "Temperature
(degrees F)")
grid(5, 5, lwd = 2) # 5 cells in both the x and y directions

help(grid)

# Example 2 Extended. Compute the average temperature and plot the temperature data. (Etter 18)
# “Use data from a physics experiment”
#
time <- c(0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0)
temps <- c(105, 126, 119, 129, 132, 128, 131, 135, 136, 132, 137)
average <- mean(temps)
plot(time, temps, type = "l", main = "Temperature Measurement", xlab = "Time (min)", ylab = "Temperature
(degrees F)", axes = FALSE)
grid(5, 5, lwd = 2) # 5 cells in both the x and y directions
axis(1, seq(0, 5, by = 0.5))
axis(2, seq(105, 140, by = 5))
box()

help(axis)
help(seq)
help(box)


library(pracma) # load pracma library

# Example 3. These statements compare linear interpolation with cubic spline interpolation (Etter 115-116)
#
x <- seq(0, 5, by = 0.5)
y <- c(72.5, 78.1, 86.4, 92.3, 110.6, 111.5, 109.3, 110.2, 110.5, 109.9, 110.2)
newx <- c(0.3, 1.25, 2.36, 4.48)
newy_1 <- interp1(x, y, newx, method = "linear")
newy_2 <- interp1(x, y, newx, method = "spline")

plot(newx, newy_1, type = "b", col = "blue", axes = FALSE, pch = 21, main = "Linear and Cubic Spline
Interpolation", xlab = "Time (s)", ylab = "Temperature (degrees F)")
lines(newx, newy_2, type = "b", col = "red", pch = 22)
legend("right", c("Linear", "Spline"), col = c("blue", "red"), pch = 21:22)
axis(1, seq(0, 5, by = 0.5))
axis(2, seq(70, 120, by = 5))
grid(5, 5, lwd = 2) # 5 cells in both the x and y directions
box()

help(interp1)
help(pch)
help(lines)
help(legend)

# Example 4. What is the value of v (velocity) for a free-falling bungee jumper? (Chapra 7-8)
#
g <- 9.81 # m/s^2
m <- 68.1 # kg
cd <- 0.25 # kg/m
t <- seq(0, 20, by = 2) # seconds / time values from 0 to 20 seconds in increments or steps of 2 seconds
#  Find v
v <- sqrt(g * m / cd) * tanh(sqrt(g * cd / m) * t) # m/s

# Example 5. Determine the drag coefficient. (Chapra 20-21)
#
m <- c(83.6, 60.2, 72.1, 91.1, 92.9, 65.3, 80.9) # kg
vt <- c(53.4, 48.5, 50.9, 55.7, 54, 47.7, 51.1) # m/s
g <- 9.81 # m/s^2
cd <- g * m / (vt ^ 2) # kg/m
```