

Introduction to R: Homework

Aron C. Eklund H. Bjørn Nielsen

October 6, 2008

1 Exercise 1

1.1

There is a really exciting data set included with the default R installation, but I can't remember what it's called. I think it has something to do with *chicken*. Find it by keyword search and load it into your workspace.

```
> help.search("chicken")
```

```
> data("chickwts")
```

1.2

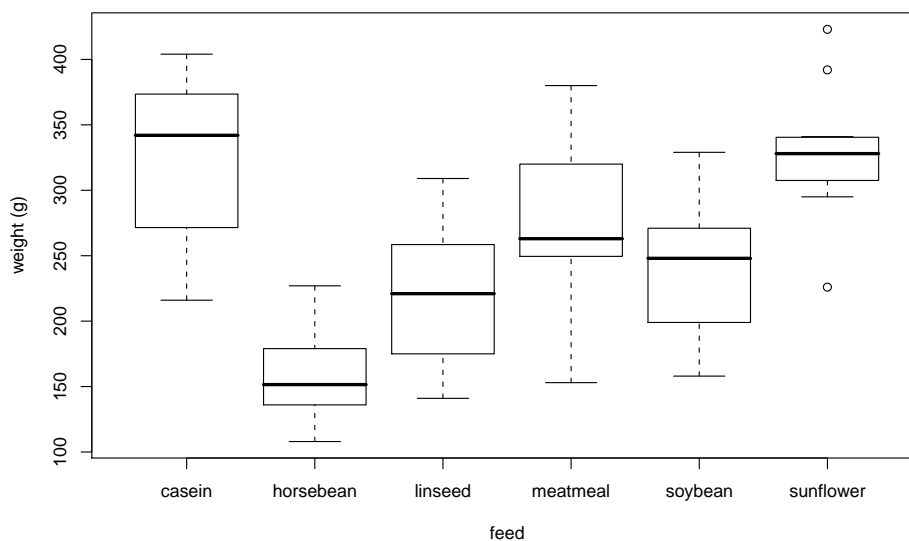
Add a new column to `chickwts` that gives the weight in kilograms, rather than grams. Make sure the column has a descriptive name.

```
> chickwts$kg <- chickwts$weight/1000
```

1.3

Make a box-and-whiskers plot showing the distribution of chicken weight according to feed type. Make sure to label the axes appropriately.

```
> boxplot(chickwts$weight ~ chickwts$feed, xlab = "feed", ylab = "weight (g)")
```



2 Exercise 2

2.1

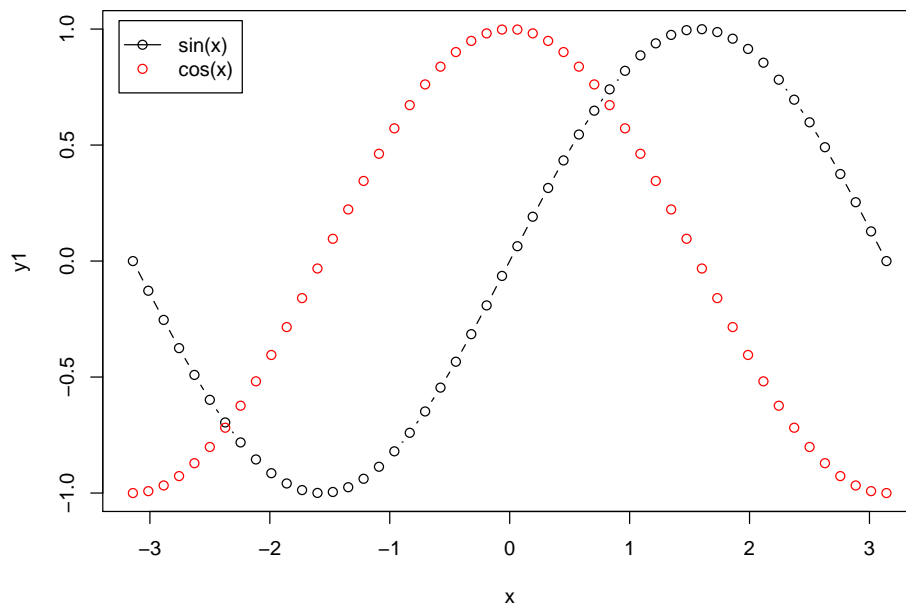
Create a numeric vector x of length 50 that ranges from $-\pi$ to π . Create a numeric vector $y1$ that is the sine of x (in radians). Create a vector $y2$ that is the cosine of x .

```
> x <- seq(-pi, pi, length = 50)
> y1 <- sin(x)
> y2 <- cos(x)
```

2.2

Plot $y1$ vs. x as a series of points joined by lines. On the same graph, add red-colored points for $y2$ vs. x . Add a legend.

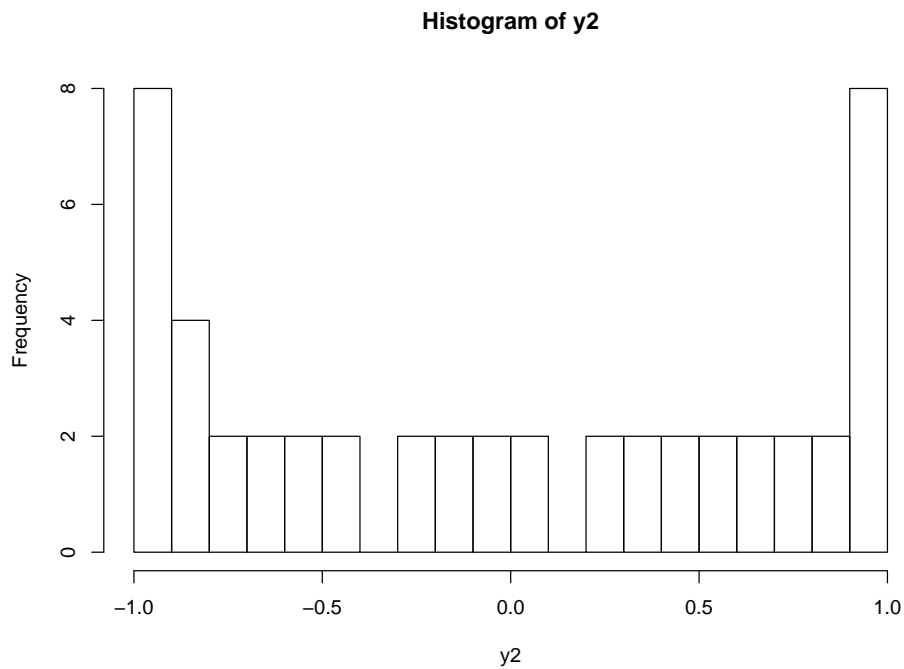
```
> plot(x, y1, type = "b")
> points(x, y2, col = "red")
> legend("topleft", legend = c("sin(x)", "cos(x)"), col = c("black",
+ "red"), pch = 1, lty = c(1, NA), inset = 0.02)
```



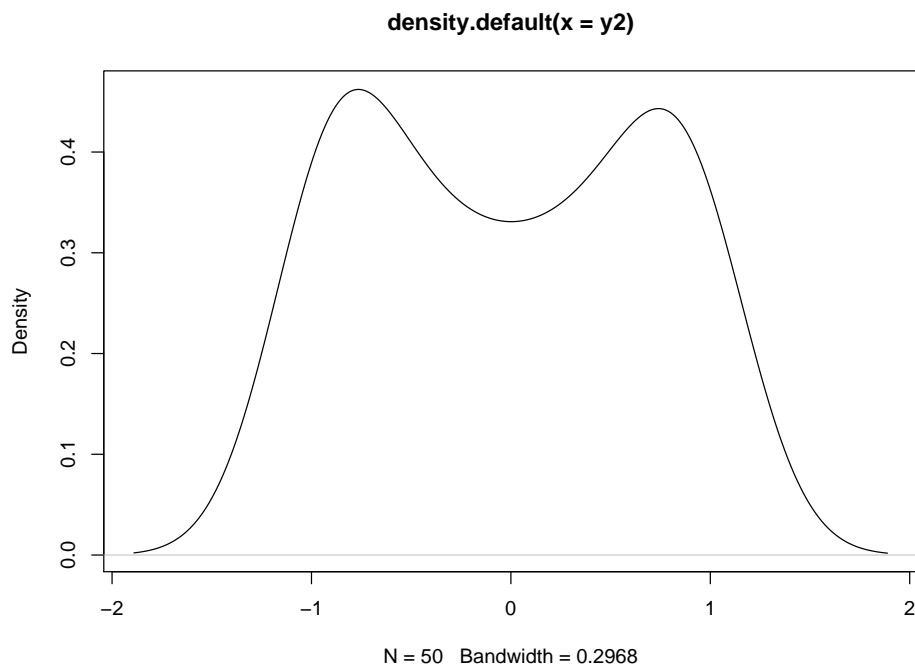
2.3

Plot a histogram of y_2 , making sure there are enough bins to clearly see the trend. Plot a density curve of y_2 using the default parameters. Which plot is more faithful to the true distribution?

```
> hist(y2, breaks = 20)
```



```
> plot(density(y2))
```



3 Exercise 3

3.1

Point your web browser at <http://data.genome.duke.edu/platinum.php>. Download the "OVC Clinical Data" spreadsheet to your local machine. Now, import the spreadsheet as a data frame into your R workspace, naming the resulting object "clin". Briefly inspect the data.

```
> library(gdata)
> clin <- read.xls("OVCclinicalinfo.xls")
```

```
Converting xls file to csv file... Done.
Reading csv file... Done.
```

```
> str(clin)
```

```
'data.frame':      119 obs. of  11 variables:
 $ OVC.TumorID      : Factor w/ 119 levels "0.08","1024",...: 1 52 53 54 2 3 4 5 6 7 ...
 $ Survival         : int  14 17 185 183 13 75 132 108 74 33 ...
 $ X0...alive...1...dead: int  1 1 0 0 1 1 1 1 1 1 ...
 $ Assigned.Stage   : int  4 4 3 3 4 3 3 3 3 3 ...
 $ GRADE            : Factor w/ 11 levels "", "0 2/3", "1",...: 7 7 7 5 7 5 7 7 5 5 ...
 $ Debulk           : Factor w/ 5 levels "0", "Optimal",...: 3 1 3 3 3 3 3 1 1 3 ...
 $ CA125.POST       : Factor w/ 88 levels "", "10", "10.5",...: 74 18 40 80 10 76 24 50 63 82 ...
 $ response.0.NR..1.CR : int  0 0 1 1 1 1 1 1 1 1 ...
 $ X                : logi  NA NA NA NA NA NA ...
 $ X.1              : logi  NA NA NA NA NA NA ...
 $ X.2              : logi  NA NA NA NA NA NA ...
```

```
> summary(clin)
```

```
      OVC.TumorID      Survival      X0...alive...1...dead Assigned.Stage
0.08 : 1  Min.    : 1.00      Min.    :0.0000      Min.    :2.000
1024 : 1  1st Qu.: 16.00     1st Qu.:0.0000     1st Qu.:3.000
1447 : 1  Median  : 34.00     Median :1.0000     Median :3.000
1451 : 1  Mean    : 49.27     Mean    :0.5798     Mean    :3.153
1504 : 1  3rd Qu.: 74.00     3rd Qu.:1.0000     3rd Qu.:3.000
1526 : 1  Max.    :185.00     Max.    :1.0000     Max.    :4.000
(Other):113
      NA's    :1.000
      GRADE      Debulk      CA125.POST response.0.NR..1.CR
2       :51  0           :42  7       : 7  Min.    :0.0000
3       :35  Optimal      :22 10       : 4  1st Qu.:0.0000
3       :20  S           :49  8       : 4  Median :1.0000
2       : 3  Suboptimal  : 5  9       : 4  Mean   :0.7143
0 2/3   : 2  Suboptimal (10/3/97): 1 11      : 3  3rd Qu.:1.0000
1       : 2                (Other):96  Max.   :1.0000
(Other): 6                NA's    : 1
      X          X.1          X.2
Mode:logical Mode:logical Mode:logical
NA's:119      NA's:119      NA's:119
```

3.2

Confirm that the last three columns are useless, and remove them. Convert the first column to `character` type. Change the name of the third column to "event". Convert the "CA125.POST" and "GRADE" columns into `numeric` values, with the ambiguous entries coded as `NA`s. Fix the "Debulk" column such that only two levels are used ("O" and "S"). Rename the eighth column to "response" and convert it to a logical vector (0 = FALSE, 1 = TRUE).

```
> clin <- clin[, 1:8]
> clin$OVC.TumorID <- as.character(clin$OVC.TumorID)
> colnames(clin)[3] <- "event"
> clin$CA125.POST <- as.numeric(as.character(clin$CA125.POST))
> clin$GRADE <- as.numeric(as.character(clin$GRADE))
> clin$Debulk <- factor(substr(clin$Debulk, 1, 1))
> colnames(clin)[8] <- "response"
> clin$response <- as.logical(clin$response)
> str(clin)

'data.frame':      119 obs. of  8 variables:
 $ OVC.TumorID   : chr  "0.08" "860" "872" "922" ...
 $ Survival      : int   14 17 185 183 13 75 132 108 74 33 ...
 $ event        : int    1 1 0 0 1 1 1 1 1 1 ...
 $ Assigned.Stage: int    4 4 3 3 4 3 3 3 3 3 ...
 $ GRADE        : num    3 3 3 2 3 2 3 3 2 2 ...
 $ Debulk       : Factor w/ 2 levels "O","S": 2 1 2 2 2 2 2 1 1 2 ...
 $ CA125.POST   : num    72.3 133 3.1 9 12 8.6 16 4 5.1 9.9 ...
 $ response     : logi  FALSE FALSE  TRUE  TRUE  TRUE  TRUE ...
```

3.3

Now that you have "cleaned up" the "clin" object, save it for later use, both as an R object ("clin.rda") and also as a CSV file ("clin.csv").

```
> save(clin, file = "clin.rda")
> write.csv(clin, file = "clin.csv")
```