

Assignment instructions

Important: You must submit your assignment through crowdmark and it must be clear and easy to read. Assignment solutions require you to show your working to receive full credit even if your final answer is correct. The assignment can be either:

- Handwritten equations that are scanned/photographed. If you choose this option ensure your work is clear and easy to read.
- Constructed using ‘R Markdown’ from ‘RStudio’ or ‘LaTeX’. ****R Markdown is strongly recommended.**** This, for example is itself an ‘R Markdown’ document.

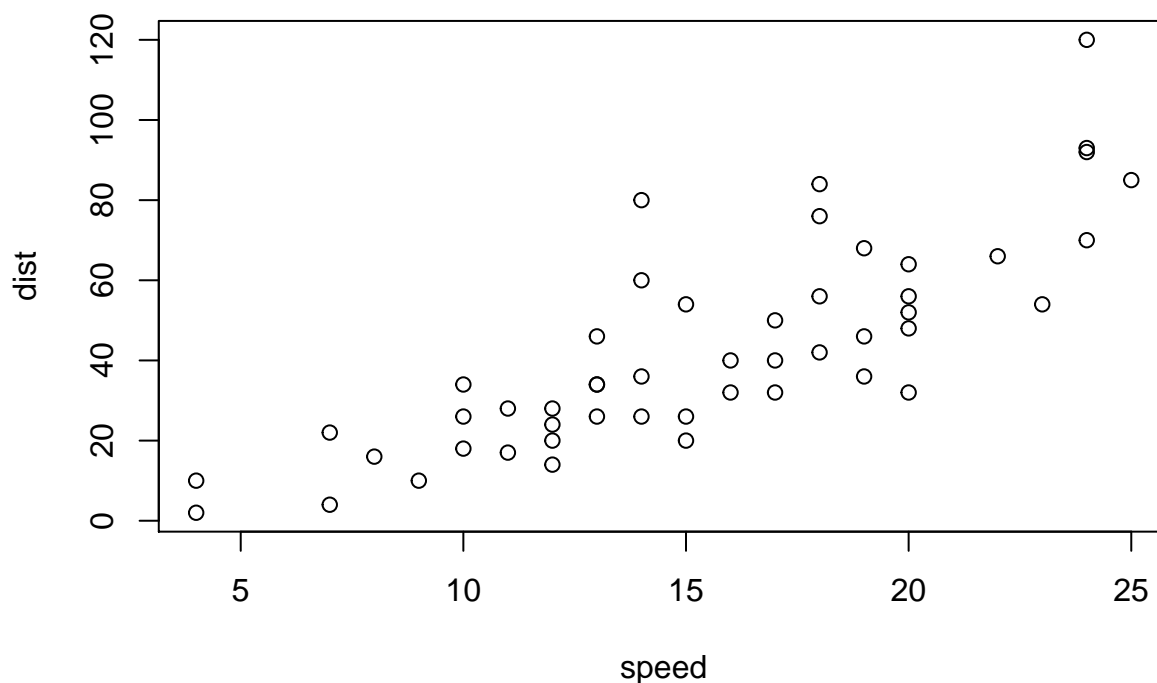
Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
## 1st Qu.:12.0    1st Qu.: 26.00
## Median :15.0    Median : 36.00
## Mean   :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
## Max.   :25.0    Max.   :120.00
```

You can also embed plots, for example:



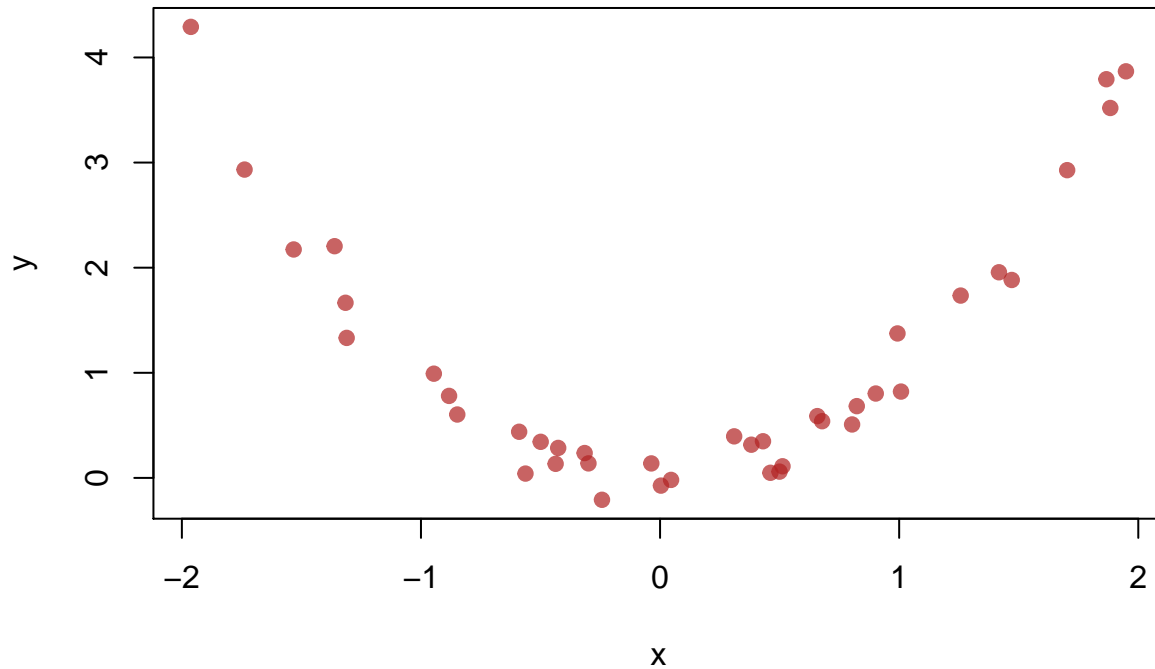
Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

In answering assignment questions, incorporate mathematics, R code, and plots as appropriate.

For example, using RMarkdown from RStudio, you might have something inline showing like this `qt(0.9, 64)` and evaluated like this 1.2949198.

You will also want to include whole chunks of code and output like this:

```
n <- 40
x <- runif(n, min=-2, max=2)
y <- x^2 + rnorm(n, sd=0.2)
plot(x,y, pch=19, col= adjustcolor("firebrick", alpha.f=0.7))
```

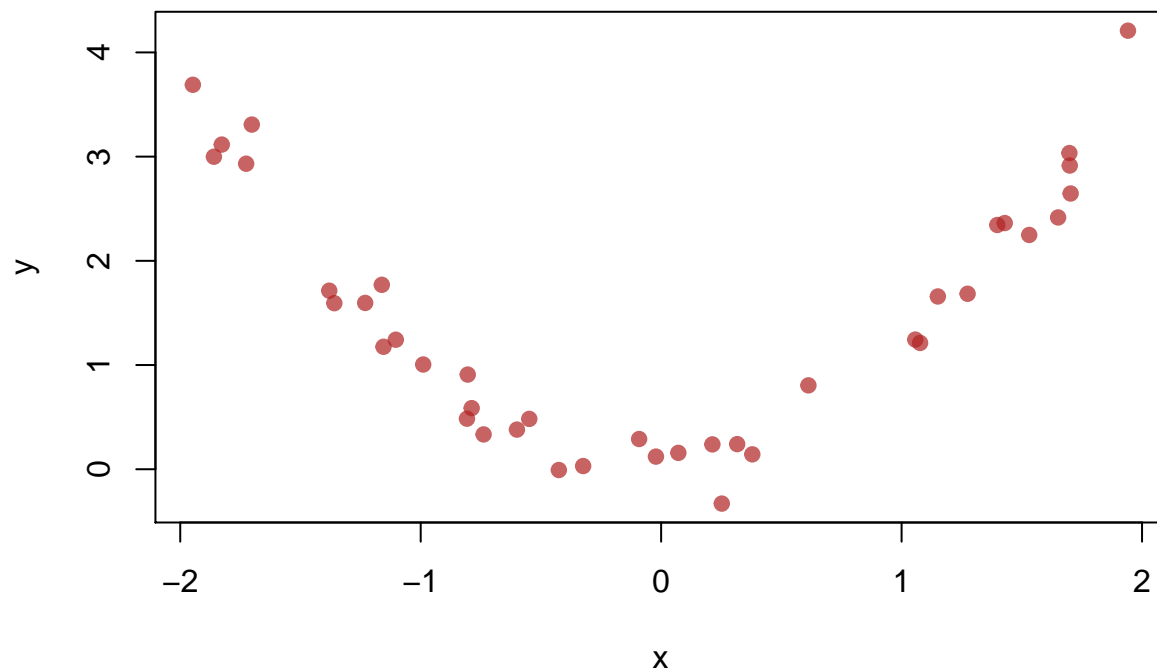


And then have your discussion appear around the code and the plot it produced.

Alternatively you might want to just show the code, without evaluating it.

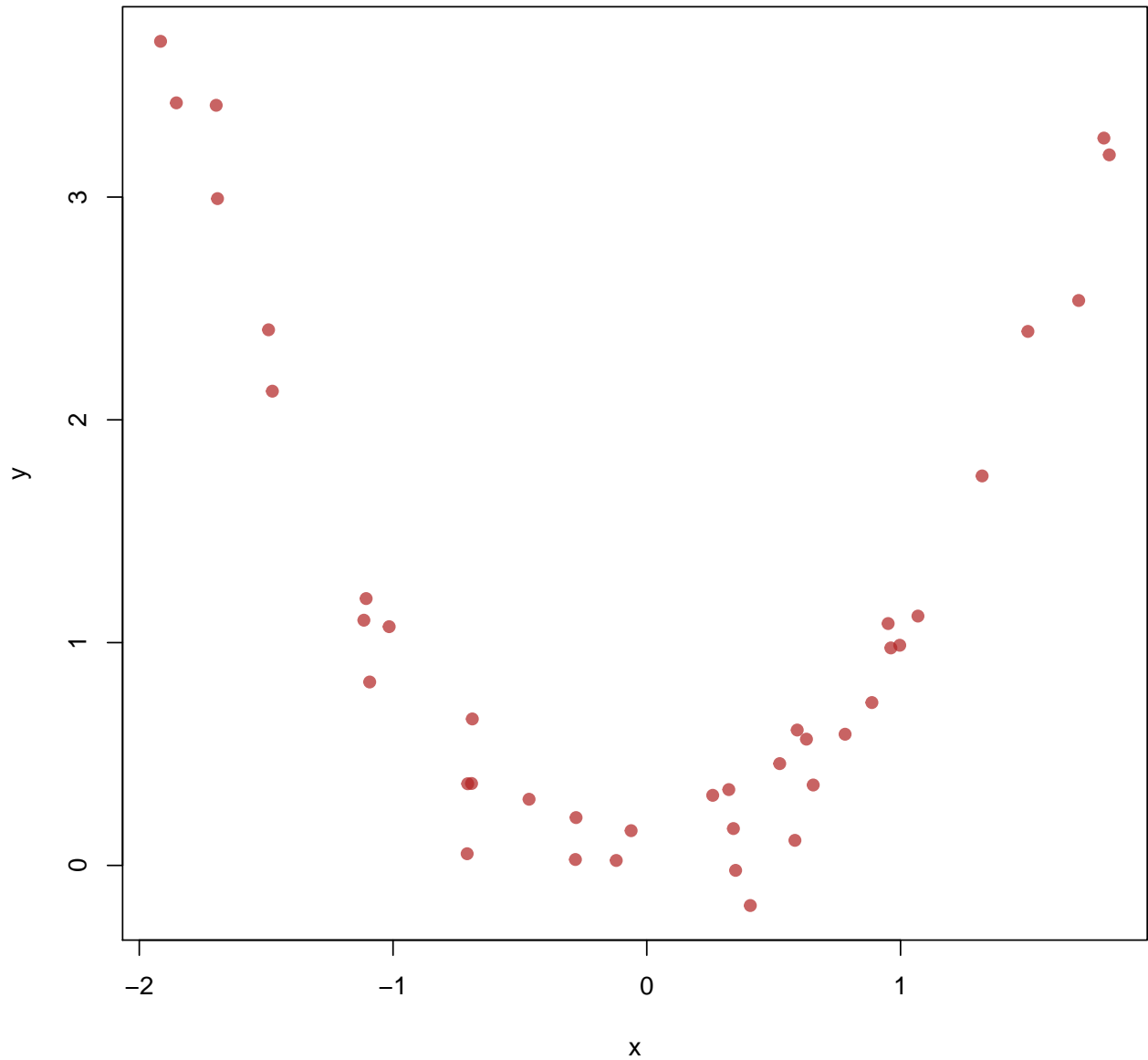
```
n <- 40
x <- runif(n, min=-2, max=2)
y <- x^2 + rnorm(n, sd=0.2)
plot(x,y, pch=19, col= adjustcolor("firebrick", alpha.f=0.7))
```

Or evaluate it but not show it



You might also want to be a bit particular about any figure you produce:

```
n <- 40
x <- runif(n, min=-2, max=2)
y <- x^2 + rnorm(n, sd=0.2)
plot(x,y, pch=19, col= adjustcolor("firebrick", alpha.f=0.7))
```



Some Math

It might, or might not, include some mathematics inline like this $\mu(\mathbf{x})$ or this $\frac{\pi}{2}$, using \$ around the texts or, using \$\$ as a separate block like this

$$Y_{new} - \tilde{\mu}(\mathbf{x}_{new}) \sim N\left(0, \sigma^2 \left(1 + \mathbf{x}_{new}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_{new}\right)\right).$$

or even a multi-equation block like:

$$\begin{aligned} \alpha &= Pr\left(-a \leq \frac{Y_{new} - \tilde{\mu}(\mathbf{x}_{new})}{\tilde{\sigma} \sqrt{1 + \mathbf{x}_{new}^T (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{x}_{new}}} \leq a\right) \\ &= Pr\left(\mathbf{I}_{new}(\mathbf{x}_{new}) \ni Y_{new}\right) \end{aligned}$$

You could also have used a different indicator for equations namely “[” and “\]”.

Here are a couple of other examples:

$$x_i = \begin{cases} 1 & \text{if } i \text{ is odd} \\ 0 & \text{otherwise.} \end{cases}$$

and

$$\mathbf{X} = \begin{bmatrix} a & b \\ c & d \\ 3 & e \end{bmatrix}$$

Make your answers complete, as if they were a report on your findings.

To include images:

To include a link

[Chris Jordan runs the numbers](#)

For miscellaneous other info, see [RStudio's RMarkdown Basics](#) or their [cheatsheet](#).

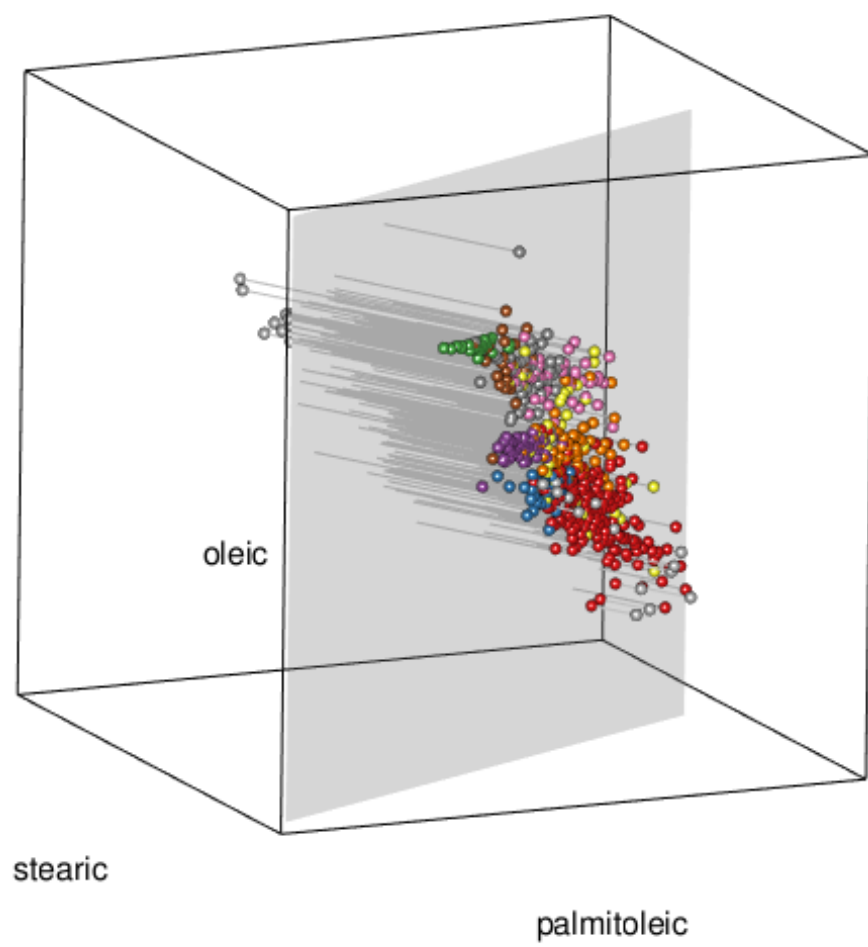


Figure 1: Moving a plane through 3d