

Solutions to R Worksheet 4.

Exercise 1

```
> x <- seq(-3,3,0.01)
> y <- exp(-abs(x))
> plot(x,y,type="l",ylab="f(x)")
> title("Plot of the double exponential function in the range [-3, 3]")
```

Exercise 2

```
> mussels <- as.data.frame(PROB3.8)
> attach(mussels)
> names(mussels) <- c("location","age","weight")

# Use logical expressions to subset the vectors

> age.l1 <- age[location==1]
> weight.l1 <- weight[location==1]
> age.l2 <- age[location==2]
> weight.l2 <- weight[location==2]

# create the output file

> plot(age.l2, weight.l2, type="b", pch="2", xlab="Age", ylab="Weight")
> lines(age.l1, weight.l1, type="b", pch="1", lty=2)
> title("Growth Characteristics of Mussels\nSouthwestern Virginia, USA")
> text(c(17,14),c(3.5,11), c("Location 1", "Location 2"))

# This plots the two sets of points and "joins the dots", which
# is a sensible approach in this instance, as the data is sorted by age
# within each location:

> reg.l1 <- lsfit(age.l1,weight.l1)
> reg.l2 <- lsfit(age.l2,weight.l2)
> plot(age.l2, weight.l2, pch="2", xlab="Age", ylab="Weight")
> points(age.l1, weight.l1, pch="1")
> abline(reg.l1$coef)
> abline(reg.l2$coef,lty=2)
> title("Growth Characteristics of Mussels\nSouthwestern Virginia, USA")
> text(c(17,14),c(3.5,11), c("Location 1", "Location 2"))

# Both locations show an increasing relationship between weight and age,
# ie. mussels increase in weight as they get older. The mussels at
# location 2 generally increase in weight faster than at location 1.
```

Exercise 3

```
> x <- rnorm(50)
> median(x)
[1] 0.08078582
> mean(x)
[1] 0.09651146

> x[1] <- x[1] + 100
> median(x)
[1] 0.09375453
> mean(x)
```

```
[1] 2.096511
```

```
# The median is only slightly affected by the addition of such a  
# distinct outlier, however the mean is definitely affected.  
# Note the above are only typical results, the results will change  
# slightly each time you repeat this simulation exercise!
```

```
> means.from.normal <- rep(0, 100)  
> medians.from.normal <- rep(0, 100)  
> means.from.cauchy <- rep(0, 100)  
> medians.from.cauchy <- rep(0, 100)  
> for(i in 1:100) {  
+   x <- rnorm(50)  
+   y <- rcauchy(50)  
+   means.from.normal[i] <- mean(x)  
+   medians.from.normal[i] <- median(x)  
+   means.from.cauchy[i] <- mean(y)  
+   medians.from.cauchy[i] <- median(y)}
```

```
# Generating histograms, means and standard deviations:
```

```
> par(mfrow=c(2,2))  
> hist(means.from.normal)  
> hist(medians.from.normal)  
> hist(means.from.cauchy)  
> hist(medians.from.cauchy)  
> mean(means.from.normal)  
[1] 0.03725064  
> mean(medians.from.normal)  
[1] 0.04554992  
> mean(means.from.cauchy)  
[1] 1.416684  
> mean(medians.from.cauchy)  
[1] -0.02197716
```

```
> median(means.from.normal)  
[1] 0.04068772  
> median(medians.from.normal)  
[1] 0.05914196  
> median(means.from.cauchy)  
[1] -0.05683651  
> median(medians.from.cauchy)  
[1] -0.01108079
```

```
> sqrt(var(means.from.normal))  
[1] 0.1403015  
> sqrt(var(medians.from.normal))  
[1] 0.1749267  
> sqrt(var(means.from.cauchy))  
[1] 10.11843  
> sqrt(var(medians.from.cauchy))  
[1] 0.2366025
```

```
# All the above show that only the means of the samples from the cauchy  
# distribution are affected by the presence of outliers in the cauchy  
# distribution. The median is robust against the presence of these outliers.
```

Exercise 4

```
# Using the "betachng()" function which was developed in Worksheet 3:
```

```

> betajck <- function(resp,pred) {
+   diff <- rep(0, length(resp))
+   n <- length(resp)
+   for(i in 1:n)) {
+     diff[i] <- betacnhg(resp,pred,i)[2]}
+   (n-1)*mean(diff)}

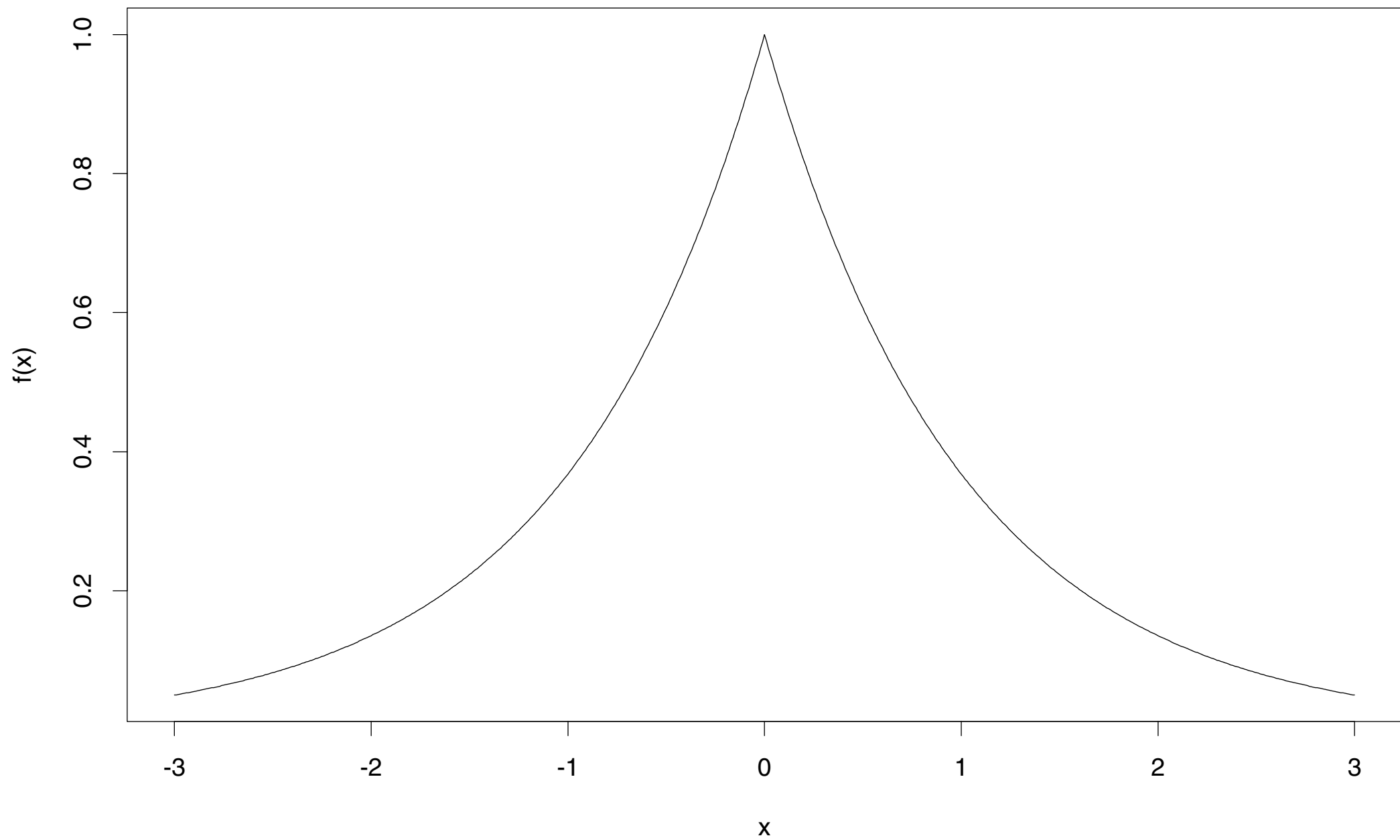
# To generalise this function, we would have to allow the user to pass in
# their estimator, "theta()", as an argument to the function, so the
# first line of the function would read:
#
# jcknfe <- function(data, theta) {
#
# Also, we would have to replace the line with the "betachng()" function by:
#
#   diff[i] <- theta(data)-theta(data[-i])
#
# So, the general jackknife function would look like:

> jcknfe <- function(data,theta) {
+   diff <- rep(0, length(data))
+   n <- length(data)
+   for(i in 1:n)) {
+     diff[i] <- theta(data)-theta(data[-i])
+   (n-1)*mean(diff)}

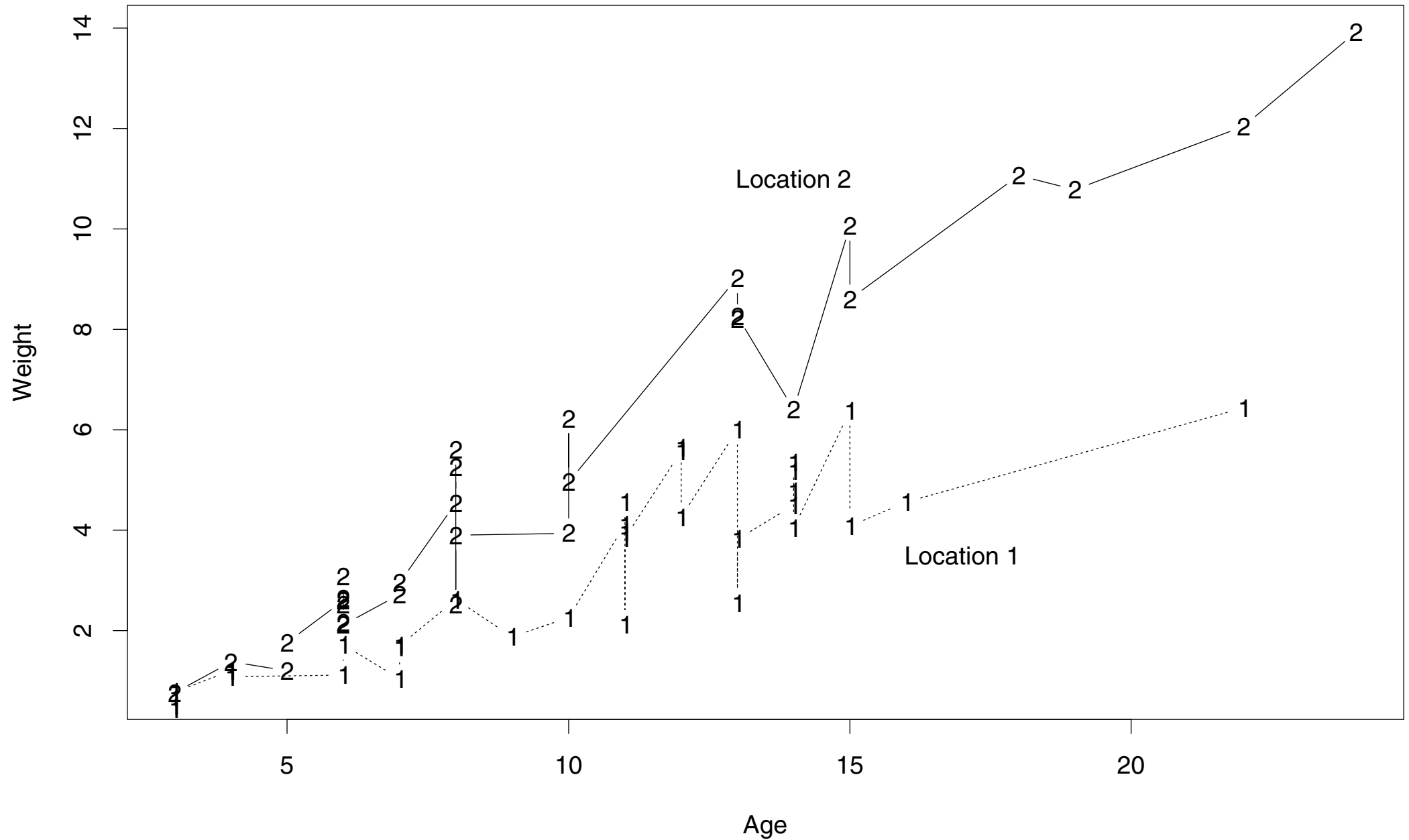
# Of course, this will only work if the input data is a vector. Try and
# think how you might modify the function even further to allow for either
# vector or matrix data input.

```

Plot of the double exponential function in the range $[-3, 3]$



Growth Characteristics of Mussels Southwestern Virginia, USA



Growth Characteristics of Mussels Southwestern Virginia, USA

