Exam 1 (part 2)

David Lewis

March 26, 2025

Question 6: Write a function se() which calculates the standard error of its argument x

```r
test_cases <- list(c(1, 2, 3, 4, 5), c(3, 5, "a", 7), c(3, NA, 8, 2))

se <- function(x) {
    sd(x) / sqrt(length(x))
}

lapply(test_cases, se)
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
```

```
## [[1]]
## [1] 0.7071068
##
## [[2]]
## [1] NA
##
## [[3]]
## [1] NA
```

As we see from the test cases, if a non-numeric value (or NA) is present in the vector, then the result of the function is NA.

Question 7: improve the definition of se() from Question 6

```r
se <- function(x, na.rm = FALSE) {
    # Check if non-numeric
    if (!is.numeric(x)) {
        warning("Argument is not numeric: Returning NA")
        return(NA)
    }
    # remove NAs
    if (na.rm) {
        x <- na.omit(x)
    }
    sd(x) / sqrt(length(x))
}
```

```r
se(c(3, 5, "a", 7))
```

```
## Warning in se(c(3, 5, "a", 7)): Argument is not numeric: Returning NA
## [1] NA
```

```r
se(c(3, NA, 8, 2))
```

```
## [1] NA
```

```r
se(c(3, NA, 8, 2), na.rm = TRUE)
```

```
## [1] 1.855921
```

Question 8

The built-in data “trees” provide the measurements of diameter (Girth), height and volume of trees. These are in inches for Girth, feet for Height and cubic feet for Volume. We will focus on Girth and Height. Write a function convert() that converts the values into the metric system (meters). Your function should take as arguments the value to convert and the unit it was expressed in (inches or feet) (use the following: x inches = x*0.0254 meters, x feet = x*0.3048 meters). If the first argument is not numeric or if the second argument is not “inch” or “feet”, use stop() function with an error message.

```r
convert <- function(x, units) {
    if (!(units == "feet" || units == "inches")) {
        stop('The unit was not "inches" or "feet"')
    }
    if (!is.numeric(x)) {
        stop("The data is not numeric")
    }
    if (units == "inches") {
        return(x * 0.0254)
    }
    # feet
    return(x * 0.3048)
}

convert(8.3, "inches")
```

```
## [1] 0.21082
```

```r
convert(8.3, "feet")
```

```
## [1] 2.52984
```

```r
convert(8.3, "foot")
```

```
## Error in convert(8.3, "foot"): The unit was not "inches" or "feet"
```

Use your function to convert the values of Girth and Height to meters. Using cor.test() function, calculate the correlation (and the corresponding p-value) between Girth and Height in meters.

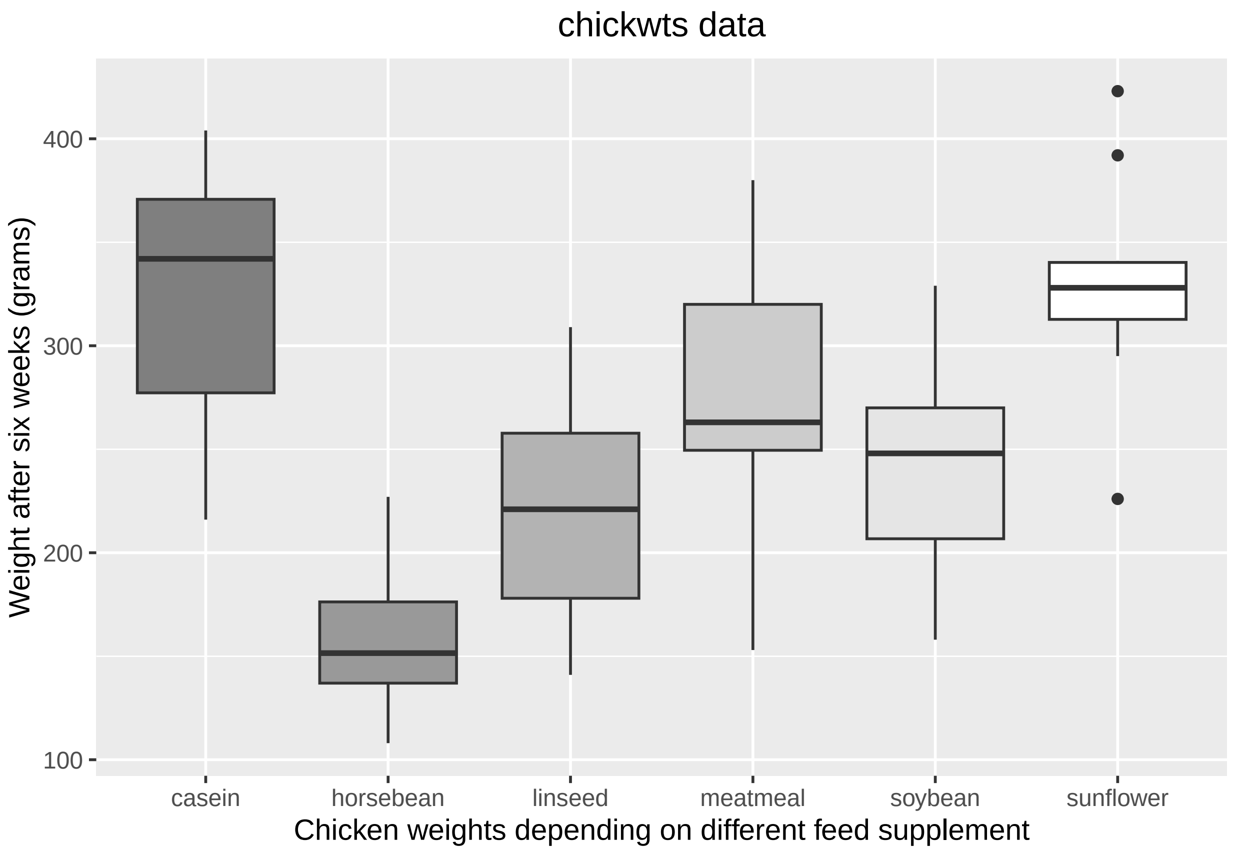
```r
Height_meters <- convert(trees$Height, "feet")
Girth_meters <- convert(trees$Girth, "inches")

cor.test(Girth_meters, Height_meters)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Girth_meters and Height_meters
## t = 3.2722, df = 29, p-value = 0.002758
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2021327 0.7378538
## sample estimates:
##       cor
## 0.5192801
```

Question 9

Using the built-in chickwts data (contains measured growth rates of chickens in dependence of various feed supplements), produce the following figure using ggplot(): Use the colors 'grey50', 'grey60', 'grey70', 'grey80', 'grey90' and 'grey100' for the bars. Note the custom titles and axis labels.

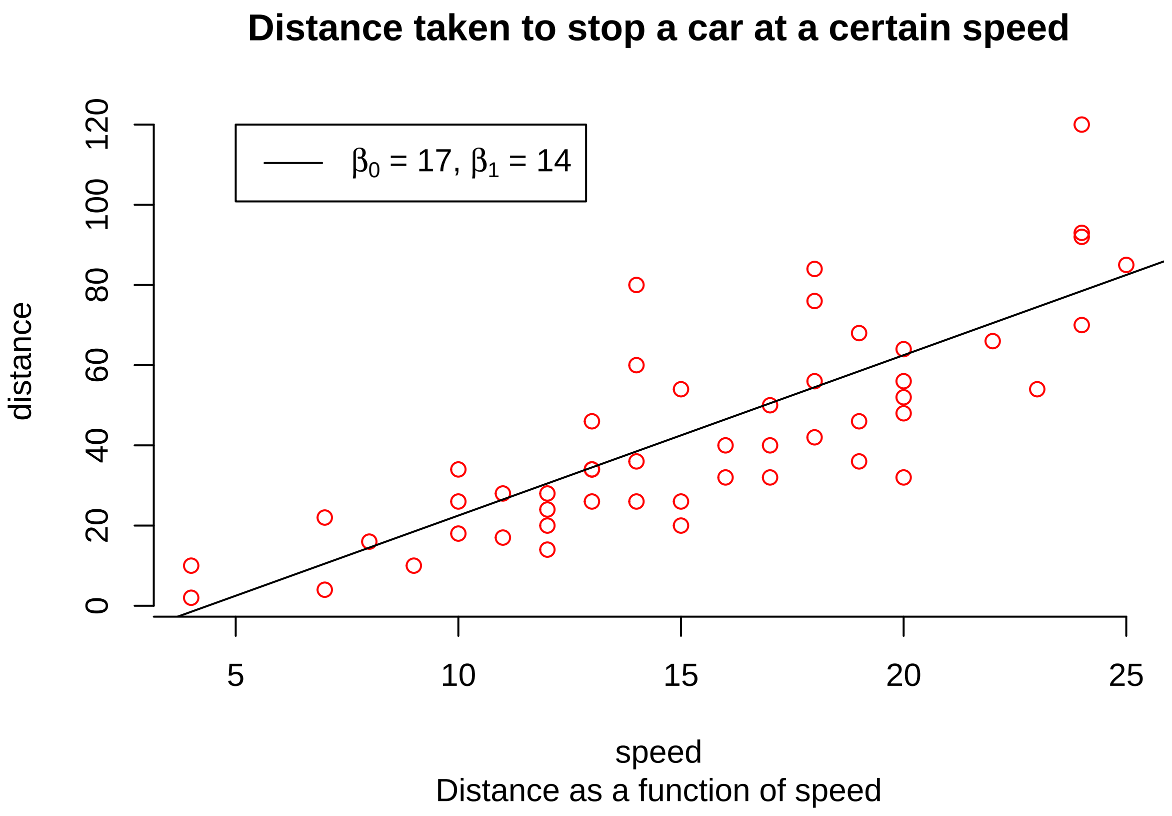
```
ggplot(chickwts, aes(x = feed, y = weight)) +
    geom_boxplot(fill = c("grey50", "grey60", "grey70", "grey80", "grey90", "grey100")) +
    xlab("Chicken weights depending on different feed supplement") +
    ylab("Weight after six weeks (grams)") +
    ggtitle("chickwts data") +
    theme(plot.title = element_text(hjust = 0.5))
```



Question 10.

Using the built-in cars data (contains distances taken to stop from a certain speed). Plot dist as a function of speed without titles and without axes (axes=FALSE). Add a line with slope 4 and intercept -17.5 (you can use abline()). Furthermore add axes with the command axis(). Specify the positions of the ticks with at = 'pretty' positions, generated with pretty(speed) and pretty(dist) (e.g. at = pretty(cars$speed)), respectively. Add a legend with legend(). The text in the legend is generated with the command expression(). Furthermore add a main title and a subtitle with the command title(main =, sub =). (The command locator(1) might help you to find a good position for the legend).

```r
plot(cars$speed, cars$dist, axes = FALSE, col = "red", xlab = "speed", ylab = "distance")
abline(-17.5, 4)
axis(1, at = pretty(cars$speed))
axis(2, at = pretty(cars$dist))
legend(
    x = 5, y = 120,
    expression(paste(beta[0], " = 17", ", ", beta[1], " = 14")), lty = 1
)
title(main = "Distance taken to stop a car at a certain speed", sub = "Distance as a function of speed"
```


Distance taken to stop a car at a certain speed
β_0 = 17, β_1 = 14
distance
speed
Distance as a function of speed