Jennifer Chu, Samnang Tep, Ryan Quigley
Section 2
Group C
MATH 267A

Problem Set 4

1. According to the help page for the function `scan()`, "a field is always delimited by an end-of-line marker unless it is quoted." Graphically, lines 1 and 2 end with the user-specified delimiter `,`, but R reads `,\n`. Thus it determines that there is an empty string between these two delimiters. Additionally, the order in which the empty strings appear in the resulting vector hints at the explanation above. The last wonder pet on line 1 is Tuck and the first wonder pet on line 2 is Ming-Ming; in the resulting vector, these strings are separated by an empty string. Similarly for Ollie and The Visitor.

2. When `header = TRUE`, R expects the first line to contain the names of the variables that are associated with each column of data. The `read.table()` argument `check.names = TRUE` by default. When set to `TRUE`, this tells R to check the names of the variables in the data to ensure that they are syntactically valid variable names, i.e, name consists of letters, numbers and the dot or underline characters and starts with a letter or the dot not followed by a number [See `?read.table`]. If the names are not syntactically valid, the `make.names` function is called, which will prepend the character "X" if necessary [See `?make.names`]. This functionality can be overwritten by setting `check.names = FALSE`; however, the user must now be careful to user backticks when selecting individual variables by name using the `$` operator.

```
caffeine.bad <- read.table("caffeine.txt", header = TRUE, check.names = FALSE)
head(caffeine.bad, n = 2)
```

```
##     0 100 200
## 1 242 248 246
## 2 245 246 248
```

```
caffeine.bad$100
```

```
## Error: <text>:1:14: unexpected numeric constant
## 1: caffeine.bad$100
##                  ^
```

```
caffeine.bad$`100`
```

```
## [1] 248 246
```

3.

c) The time that it took to compute $\hat{\beta}$:

```
##    user  system elapsed
##   2.087   0.032   2.128
```

d) The time that it took to compute $\hat{\beta}$:

```
##    user  system elapsed
##   1.558   0.012   1.577
```

e) The formula for $\hat{\beta}$ can be rearranged in the following way,

$$\begin{aligned}
\hat{\beta} &= (X^T X)^{-1} X^T y \\
(X^T X)\hat{\beta} &= (X^T X)(X^T X)^{-1} X^T y \\
(X^T X)\hat{\beta} &= I X^T y \\
(X^T X)\hat{\beta} &= X^T y
\end{aligned}$$

The time that it took to compute $\hat{\beta}$:

```
##    user  system elapsed
##   0.827   0.007   0.836
```

g) TODO: The computation time decreases with each but why???

h) Once `X` has been converted from a matrix to a data frame, $\hat{\beta}$ cannot be calculated using the code from part (c) because the matrix multiplication operator `%*%` does not work with data frames. Data frames have the ability to store values of different types, which is not appropriate for matrix multiplication. Instead of checking all of the columns of the data frame to makes sure they are all numeric (or complex), R simply checks the class of the objects being multiplied and throws an error if they are not appropriate.

```
## Error in x.t %*% X.df: requires numeric/complex matrix/vector arguments
```

```
## Timing stopped at: 0.073 0.01 0.083
```