p1805_hw1_zo2168

Problem 1

```
data("penguins", package = "palmerpenguins")
#Get the column names to find the name of variables
names(penguins)
```

```
## [1] "species"           "island"            "bill_length_mm"
## [4] "bill_depth_mm"     "flipper_length_mm" "body_mass_g"
## [7] "sex"               "year"
```

```
#Find unique penguin species
distinct(penguins, species)
```

```
## # A tibble: 3 x 1
##   species
##   <fct>
## 1 Adelie
## 2 Gentoo
## 3 Chinstrap
```

```
#Find the unique islands
distinct(penguins,island)
```

```
## # A tibble: 3 x 1
##   island
##   <fct>
## 1 Torgersen
## 2 Biscoe
## 3 Dream
```

```
#Find the maximum and minimum of each quantitative variables.
max(pull(penguins, bill_length_mm), na.rm = T)
```

```
## [1] 59.6
```

```
min(pull(penguins, bill_length_mm), na.rm = T)
```

```
## [1] 32.1
```

```r
max(pull(penguins, bill_depth_mm), na.rm = T)
```

```
## [1] 21.5
```

```r
min(pull(penguins, bill_depth_mm), na.rm = T)
```

```
## [1] 13.1
```

```r
max(pull(penguins, flipper_length_mm), na.rm = T)
```

```
## [1] 231
```

```r
min(pull(penguins, flipper_length_mm), na.rm = T)
```

```
## [1] 172
```

```r
max(pull(penguins, body_mass_g), na.rm = T)
```

```
## [1] 6300
```

```r
min(pull(penguins, body_mass_g), na.rm = T)
```

```
## [1] 2700
```

```r
#Find the year range
range(pull(penguins, year),na.rm = T)
```

```
## [1] 2007 2009
```

```r
#Find the size of dataset
nrow(penguins)
```

```
## [1] 344
```

```r
ncol(penguins)
```

```
## [1] 8
```

```r
mean(pull(penguins, flipper_length_mm), na.rm = T)
```

```
## [1] 200.9152
```

The variable names include species,island,bill_length_mm,bill_depth_mm,flipper_length_mm,body_mass_g, sex, year.
Three species are *Adelie*, *Gentoo*, *Chinstrap*,
Three island are *Torgersen*, *Biscoe*, *Dream*,
The maximum and minimum of bill_length_mm are 59.6 mm and 32.1 mm respectively
The maximum and minimum of bill_depth_mm are 21.5 mm and 13.1 mm respectively
The maximum and minimum of flipper_length_mm are 231 mm and 172 mm respectively
The maximum and minimum of body_mass_g are 6300 and 2700 respectively
Year ranges from 2007 to 2009
sex includes male and female
This dataset has 344 rows and 8 columns
Mean flipper length is 200.9152

```r
flipper_vs_bill <- ggplot(penguins, aes(x = bill_length_mm, y = flipper_length_mm, color = species)) +
  geom_point() +
  labs(x = 'Bill Length (mm)', y = 'Flipper Length (mm)', title = 'Flipper Length vs Bill Length by Spe
ggsave("flipper_vs_bill_scatterplot.png", plot = flipper_vs_bill, width = 6, height = 4)
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_point()‘).
```

Problem 2

```r
set.seed(1)
# Create a random sample of size 10 from a standard Normal distribution
std <- rnorm(10)
is.numeric(std)
```

```
## [1] TRUE
```

```r
# Create a logical vector indicating whether elements of the sample are greater than 0
logi <- std > 0
is.logical(logi)
```

```
## [1] TRUE
```

```r
# Create a character vector of length 10
character_vector <- c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j")
is.character(character_vector)
```

```
## [1] TRUE
```

```r
# Create a factor vector of length 10, with 3 different factor "levels"
factor_index<- sample(0:2, 10, replace = TRUE)
factor_three_levels <- factor(factor_index, labels = c("low","medium","high"))
is.factor(factor_three_levels)
```

```
## [1] TRUE
```

```
tibble(std,logi,character_vector,factor_three_levels)
```

```
## # A tibble: 10 x 4
##       std logi  character_vector factor_three_levels
##     <dbl> <lgl> <chr>            <fct>
##  1 -0.626 FALSE a                low
##  2  0.184 TRUE  b                high
##  3 -0.836 FALSE c                low
##  4  1.60  TRUE  d                low
##  5  0.330 TRUE  e                low
##  6 -0.820 FALSE f                low
##  7  0.487 TRUE  g                medium
##  8  0.738 TRUE  h                low
##  9  0.576 TRUE  i                low
## 10 -0.305 FALSE j                medium
```

```
mean(std)
```

```
## [1] 0.1322028
```

```
mean(logi)
```

```
## [1] 0.6
```

```
mean(character_vector)
```

```
## Warning in mean.default(character_vector): argument is not numeric or logical:
## returning NA
```

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

```
mean(factor_three_levels)
```

```
## Warning in mean.default(factor_three_levels): argument is not numeric or
## logical: returning NA
```

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

The `charactor_vector` and `factor_three_levels` does not have the mean.

```
as.numeric(logi)
```

```
##  [1] 0 1 0 1 1 0 1 1 1 0
```

```
as.numeric(character_vector)
```

```
## Warning: NAs introduced by coercion
```

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

```r
as.numeric((factor_three_levels))
```

```
##  [1] 1 3 1 1 1 1 2 1 1 2
```

Logical values are converted to 0 or 1, and this is true as it can be transfered to 0 or 1 meaning false or true repectively. | *Character* values are converted to NA. | Factor values are converted to 1,2, or 3, each stands for one level.| The reason could be the logical varaibles are enconded as 0 representing False and 1 representing True. | Character variables cannot be averaged because it usually contain text, and there's no numerical interpretation of strings for mean calculation. | Factor values are internally ctores as numeric values, but the calculating the mean doesn't make sense in the same way it does for numeric variables.