Chapter 2: Distributions

Ghislain Nono Gueye, Ph.D.

3/7/2019

Contents

Load useful package and import data

The data used in his chapter is the `2002FemPreg.Rds` data set

```
library(here)
library(dplyr)
library(ggplot2)
library(forcats)

fempreg2002 <- readRDS(here("data", "processed", "used-in-book", "2002FemPreg.Rds"))
```

1 Histograms

The `table()` function in `R` computes frequencies and its output is a *named vector*. It helps to convert the output into a data frame for analysis.

2 Representing histograms

```
# Similar to the Hist constructor
table_to_df <- function(x){
    df <- as.data.frame(table(x))
    colnames(df) <- c("value", "frequency")
    df$value <- as.numeric(as.character(df$value))
    df
}

# Similar to the Freq method
find_freq <- function(x, v){
    sum(x == v)
}

x <- c(1, 2, 2, 3, 5)
table_to_df(x)
```

```
##   value frequency
## 1     1         1
## 2     2         2
## 3     3         1
## 4     5         1
```

```
find_freq(x, 2)
```

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

```
find_freq(x, 4)
```

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

The `unique()` function in `R` serves the same purpose as the `Values()` method and the resulting vector contains sorted values.

```
unique(x)
```

```
## [1] 1 2 3 5
```

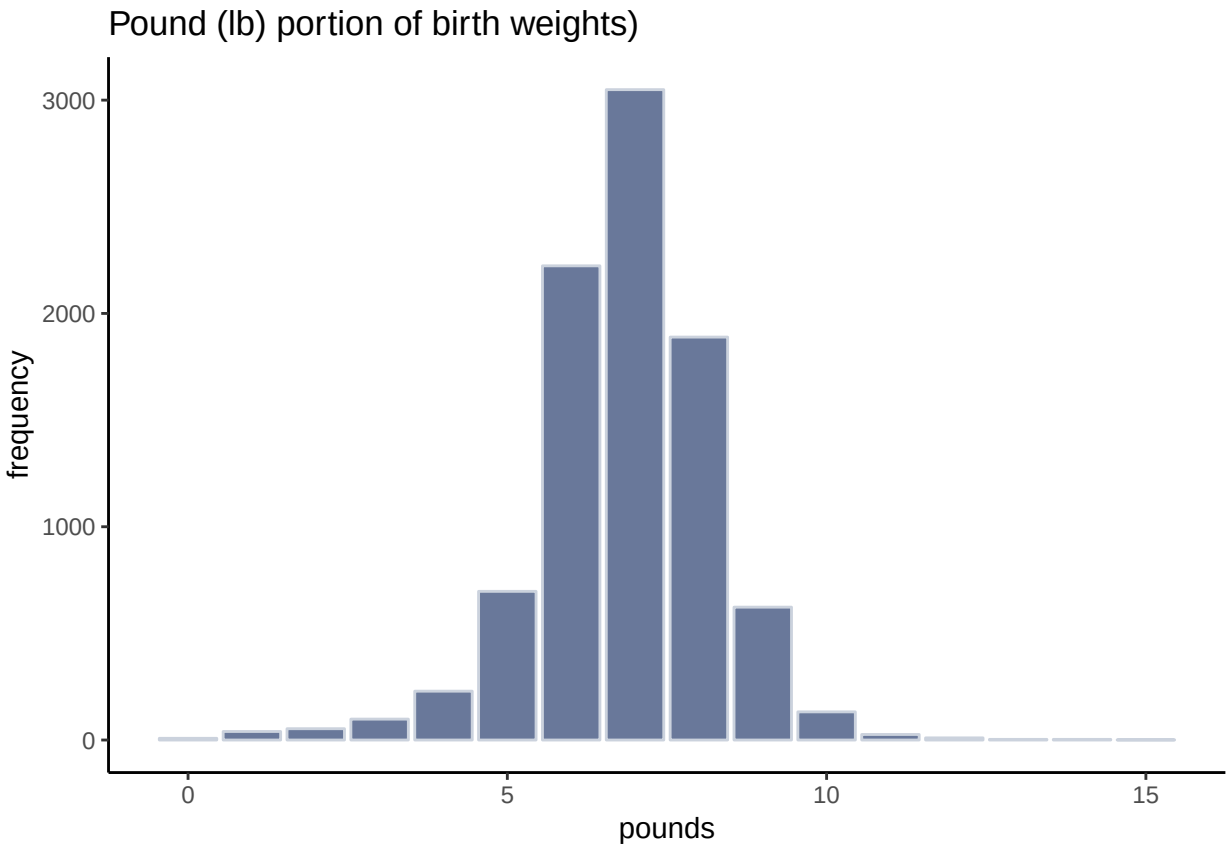
3 Plotting histograms

The `hist()` function is used to plot histograms. I personally prefer using `ggplot2` for anything pplot related.

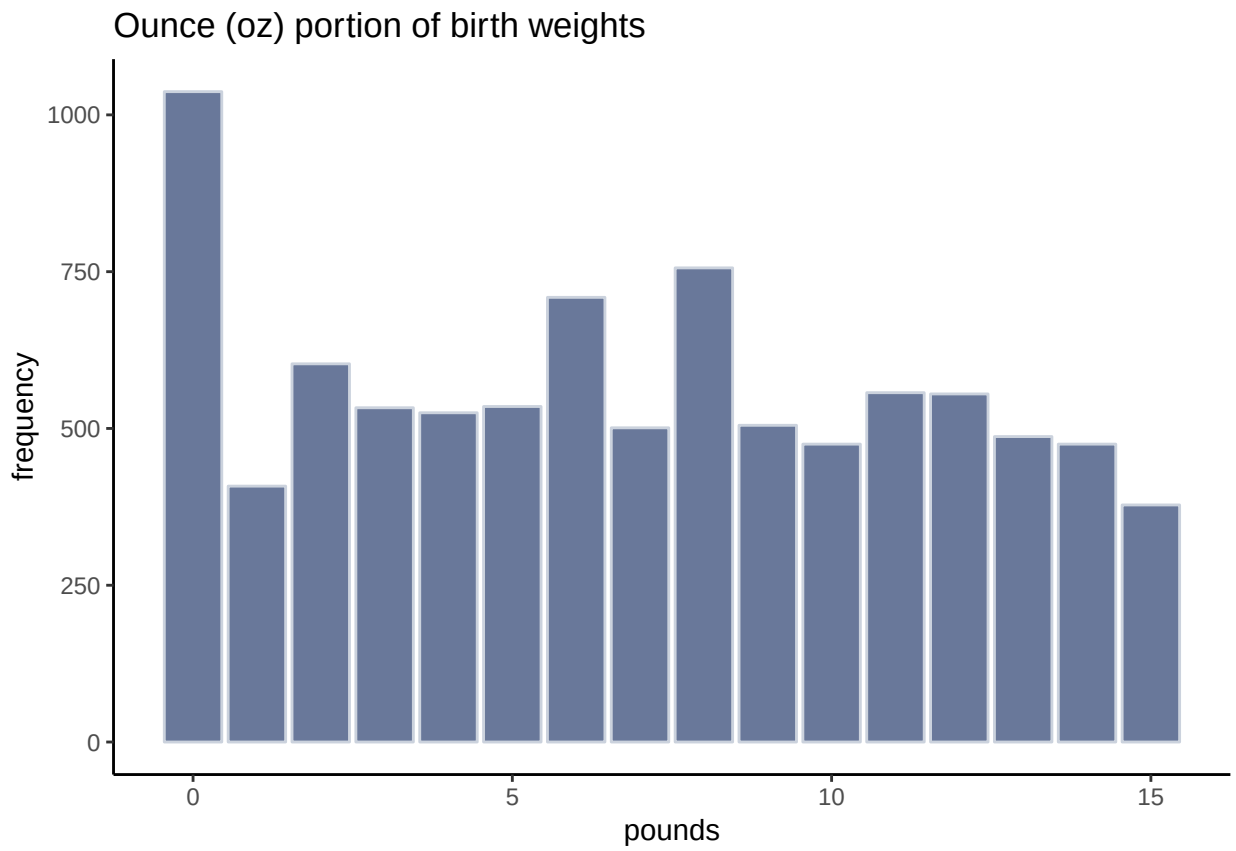
4 NSFG variables

Selecting records of live births

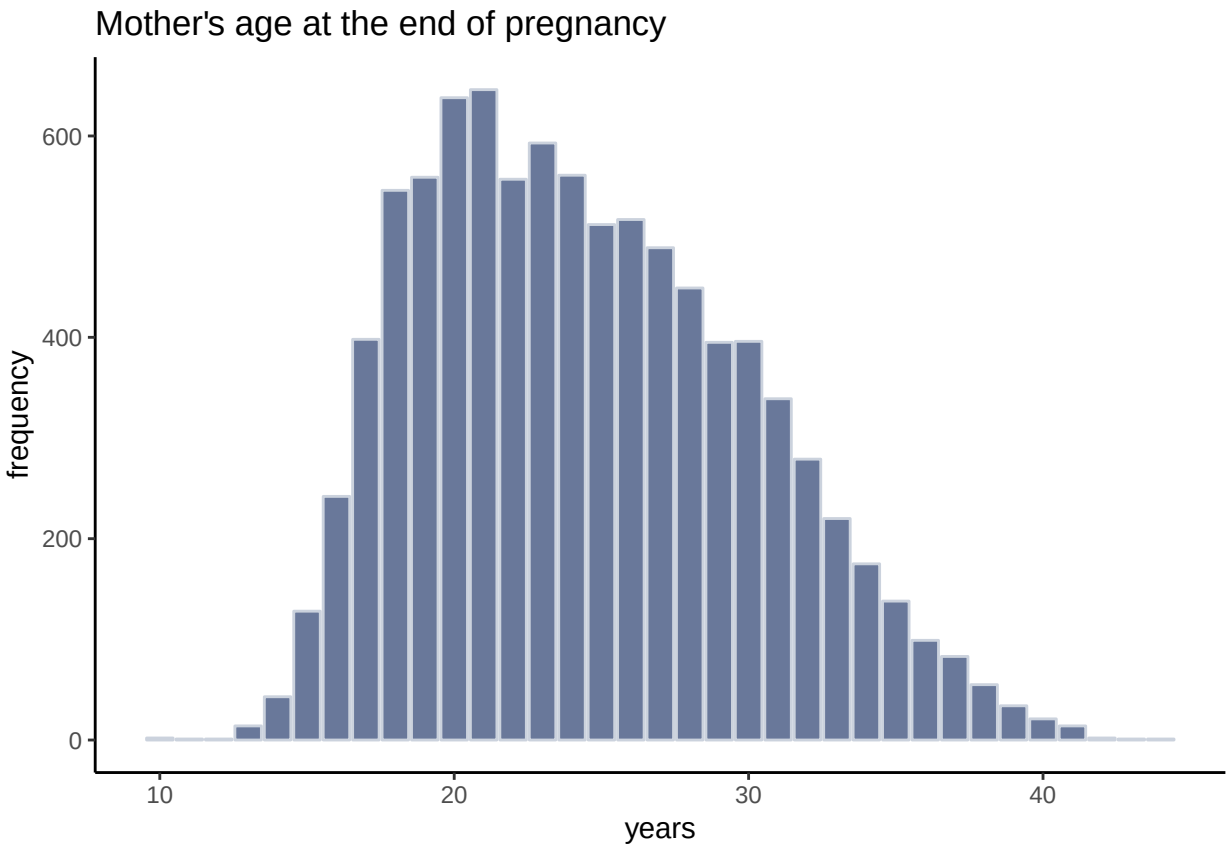
```
live_births_data <- fempreg2002 %>%
    filter(outcome == 1)

birthwgt_lb_freq <- table_to_df(live_births_data$birthwgt_lb)

ggplot(data = birthwgt_lb_freq, aes(x = value, y = frequency)) +
    geom_col(fill = "#69789A", col = "#CBD2DD") +
    theme_classic() +
    labs(x = "pounds", title = "Pound (lb) portion of birth weights)")
```



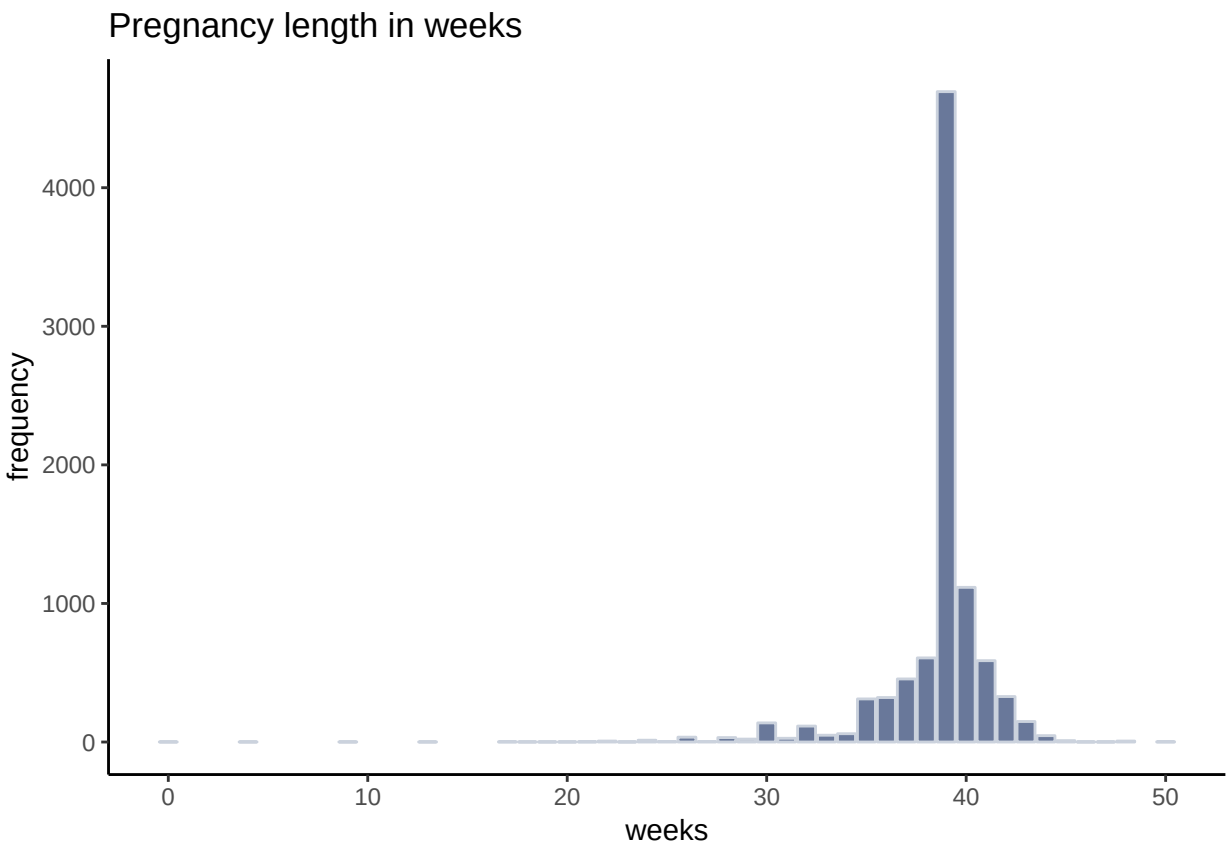

```
birthwgt_oz_freq <- table_to_df(live_births_data$birthwgt_oz)

ggplot(data = birthwgt_oz_freq, aes(x = value, y = frequency)) +
    geom_col(fill = "#69789A", col = "#CBD2DD") +
    theme_classic() +
    labs(x = "pounds", title = "Ounce (oz) portion of birth weights")
```




```
agepreg_freq <- table_to_df(as.integer(live_births_data$agepreg))

ggplot(data = agepreg_freq, aes(x = value, y = frequency)) +
    geom_col(fill = "#69789A", col = "#CBD2DD") +
    theme_classic() +
    labs(x = "years", title = "Mother's age at the end of pregnancy")
```




```
prglngth_freq <- table_to_df(live_births_data$prglngth)

ggplot(data = prglngth_freq, aes(x = value, y = frequency)) +
    geom_col(fill = "#69789A", col = "#CBD2DD") +
    theme_classic() +
    labs(x = "weeks", title = "Pregnancy length in weeks")
```



5 Outliers

```
min_n <- function(x, n){
    sort(unique(x))[1:n]
}

max_n <- function(x, n){
    sort(unique(x), decreasing = TRUE)[1:n]
}
```

```
min_n(live_births_data$prglngth, 10)
```

```
##  [1]  0  4  9 13 17 18 19 20 21 22
```

```
table_to_df(live_births_data$prglngth) %>%
    filter(value >= 43)
```

```
##   value frequency
## 1    43       148
## 2    44        46
## 3    45        10
## 4    46         1
## 5    47         1
## 6    48         7
## 7    50         2
```

6 First babies

```
first_or_not <- live_births_data %>%
    filter(prglngth >= 27 & prglngth <=46) %>%
    mutate(first_baby = factor(if_else(birthord == 1, "first", "other"))) %>%
    group_by(prglngth, first_baby) %>%
    summarize(count = n())

ggplot(data = first_or_not, aes(x = prglngth, y = count, fill = first_baby)) +
    geom_bar(position = "dodge", stat = "identity") +
    labs(fill = "baby order") +
    theme_classic()
```