Introduction to Statistical Inference (QTM 100 Lab)

Lecture 2: Summarizing and Visualizing Data

Justin Eloriaga — Emory University

Fall 2024

Gameplan

Importing the Data

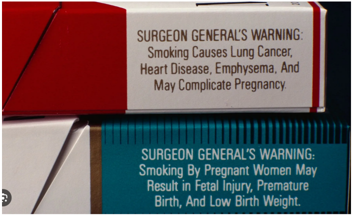
Variable Types

Numeric Variables

Categorical Variables

Importing the Data

Child Health and Development Studies


SURGEON GENERAL'S WARNING:
Smoking Causes Lung Cancer, Heart Disease, Emphysema, And May Complicate Pregnancy.
SURGEON GENERAL'S WARNING:
Smoking By Pregnant Women May Result in Fetal Injury, Premature Birth, And Low Birth Weight.

- Prior to the placement of the warning, studies had to be conducted to investigate the effects of smoking during pregnancy.
- Despite the warnings that went into effect in 1985, the National Center for Health Statistics found that 15% of women who gave birth in 1996 smoked during pregnancy.
- Why do we care about baby birth weight? Birth weight is a measure of a baby's maturity! Has consequences on future health outcomes.

Variables Under Consideration

variable	description
`bwt`	baby's weight at birth in ounces
`gestation`	length of pregnancy in days
`parity`	0=first born, 1=otherwise
`age`	mother's age in years
`height`	mother's height in inches
`weight`	mother's pregnancy weight in pounds
`smoke`	smoking status of mother: 0=not now, 1=yes now

Importing the Dataset (again)

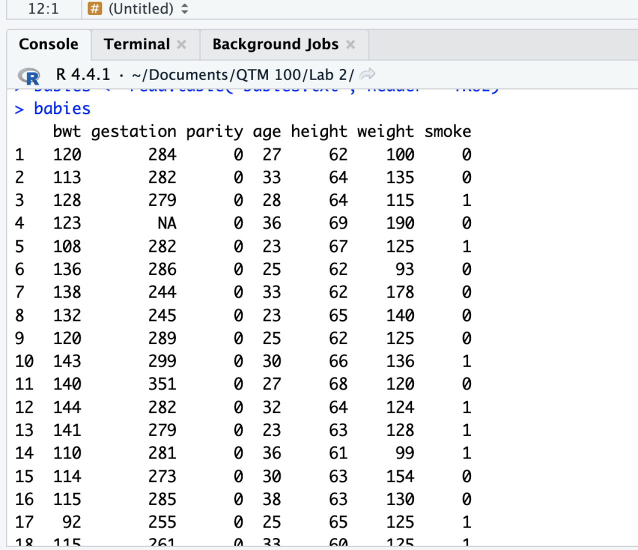
- Like before, we can use point-and-click or the working directory

```
setwd("YourFilePath")
```

```
babies <- read.table("babies.txt", header = TRUE)
```

- The only difference from before is that the file extension is `.txt`, so we must use `read.table()` to import that.

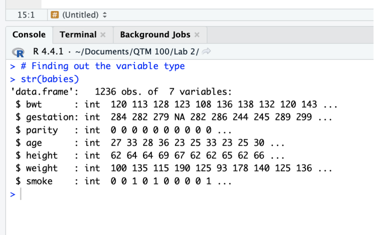
The Dataset, at a glance



```
> babies <- read.table("babies.txt", header = TRUE)
> babies
   bwt gestation parity age height weight smoke
1  120       284      0  27     62    100     0
2  113       282      0  33     64    135     0
3  128       279      0  28     64    115     1
4  123        NA      0  36     69    190     0
5  108       282      0  23     67    125     1
6  136       286      0  25     62     93     0
7  138       244      0  33     62    178     0
8  132       245      0  23     65    140     0
9  120       289      0  25     62    125     0
10 143       299      0  30     66    136     1
11 140       351      0  27     68    120     0
12 144       282      0  32     64    124     1
13 141       279      0  23     63    128     1
14 110       281      0  36     61     99     1
15 114       273      0  30     63    154     0
16 115       285      0  38     63    130     0
17  92       255      0  25     65    125     1
18 115       261      0  33     60    125     1
```

Variable Types

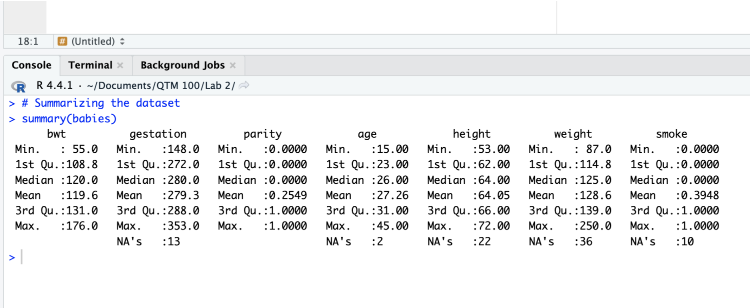
Finding out the type



```
> # Finding out the variable type
> str(babies)
'data.frame':	1236 obs. of  7 variables:
 $ bwt      : int  120 113 128 123 108 136 138 132 120 143 ...
 $ gestation: int  284 282 279 NA 282 286 244 245 289 299 ...
 $ parity   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ age      : int  27 33 28 36 23 25 33 23 25 30 ...
 $ height   : int  62 64 64 69 67 62 62 65 62 66 ...
 $ weight   : int  100 135 115 190 125 93 178 140 125 136 ...
 $ smoke    : int  0 0 1 0 1 0 0 0 0 1 ...
> 
```

- If you use the `str()` function, it tells you the type.
- For this dataset, it tells us all of the variables are of type `int` (i.e. integers or whole numbers).
- What do the `NA`'s mean?

Summarizing the Dataset



```
> # Summarizing the dataset
> summary(babies)
      bwt          gestation         parity            age            height          weight          smoke       
 Min.   : 55.0   Min.   :148.0   Min.   :0.0000   Min.   :15.00   Min.   :53.00   Min.   : 87.0   Min.   :0.0000  
 1st Qu.:108.8   1st Qu.:272.0   1st Qu.:0.0000   1st Qu.:23.00   1st Qu.:62.00   1st Qu.:114.8   1st Qu.:0.0000  
 Median :120.0   Median :280.0   Median :0.0000   Median :26.00   Median :64.00   Median :125.0   Median :0.0000  
 Mean   :119.6   Mean   :279.3   Mean   :0.2549   Mean   :27.26   Mean   :64.05   Mean   :128.6   Mean   :0.3948  
 3rd Qu.:131.0   3rd Qu.:288.0   3rd Qu.:1.0000   3rd Qu.:31.00   3rd Qu.:66.00   3rd Qu.:139.0   3rd Qu.:1.0000  
 Max.   :176.0   Max.   :353.0   Max.   :1.0000   Max.   :45.00   Max.   :72.00   Max.   :250.0   Max.   :1.0000  
                 NA's   :13                       NA's   :2       NA's   :22      NA's   :36      NA's   :10      
> 
```

We can use the summary() command we learned before to get an overview of the dataset.

Recoding Variables

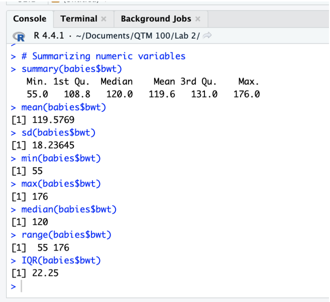
- Notice that `smoke` and `parity` variables do not represent numeric measurements!
- Although the data values are stored as 0's and 1's, in reality, these 0's and 1's represent categories.
- We would like to know how many mothers were smoking.
- For `R` to treat these appropriately as categorical variables, we need to **recode** them as factor variables

```
babies$parityf <- factor(babies$parity, labels = c("first born","otherwise"))

babies$smokef <- factor(babies$smoke, labels = c("not now","yes now"))
```

Numeric Variables

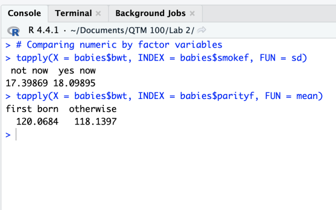
Summarizing Numeric Variables



```
> # Summarizing numeric variables
> summary(babies$bwt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   55.0   108.8   120.0   119.6   131.0   176.0 
> mean(babies$bwt)
[1] 119.5769
> sd(babies$bwt)
[1] 18.23645
> min(babies$bwt)
[1] 55
> max(babies$bwt)
[1] 176
> median(babies$bwt)
[1] 120
> range(babies$bwt)
[1]  55 176
> IQR(babies$bwt)
[1] 22.25
> 
```

- As before, we can use various commands to get a general overview of the dataset. Let us try the following commands:
    - `summary()`
    - `mean()` - Mean
    - `sd()` - Standard Deviation
    - `min()` - Minimum Value
    - `max()` - Maximum Value
    - `median()` - Median
    - `range()` - Range (Max - Min)
    - `IQR()` - Interquartile Range

Comparing Numerical Variables by Factor/Category



```
> # Comparing numeric by factor variables
> tapply(X = babies$bwt, INDEX = babies$smokef, FUN = sd)
 not now  yes now 
17.39869 18.09895 
> tapply(X = babies$bwt, INDEX = babies$parityf, FUN = mean)
first born  otherwise 
  120.0684   118.1397 
> 
```

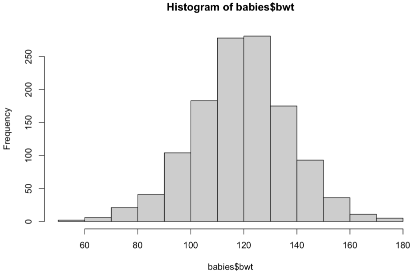
- We can use the `tapply()` function to compare some statistics of a numerical variable by factor/category.

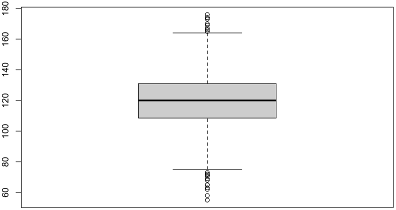
```
tapply(X = babies$bwt,INDEX = babies$smokef, FUN = sd)

tapply(X = babies$bwt,INDEX = babies$parityf, FUN = mean)
```

Visualizing Numeric Variables

We can use the `hist()` and `boxplot()` to create a histogram/boxplot respectively



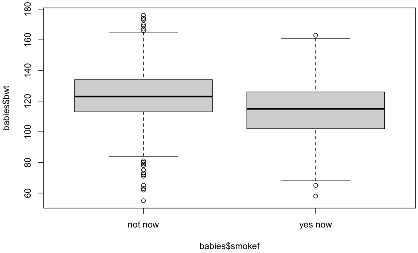


`hist(babies$bwt)`

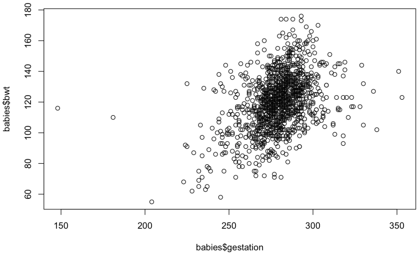
`boxplot(babies$bwt)`

Visualizing Numeric Variables

Side-by-side boxplots are commonly used to visualize the distribution of a numeric variable by groups. Scatterplots are used to visualize the distribution of two numeric variables.



```
boxplot(babies$bwt
babies$smokef)
```



```
plot(x = babies$gestation, y
= babies$bwt)
```

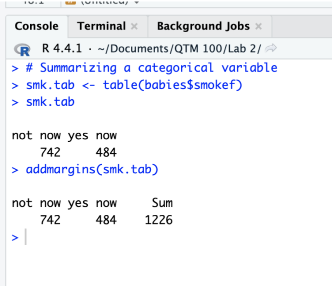
Categorical Variables

Using the `table()` function

When summarizing categorical variables, it is interesting to know the frequency of occurrences of each level of the categorical variable. We use the `table()` function in conjunction with the `addmargins()` function

```
smk.tab <- table(babies$smokef)

addmargins(smk.tab)
```



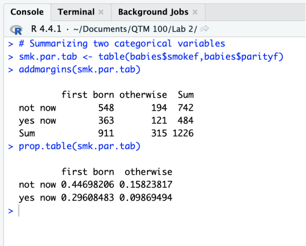
```
> # Summarizing a categorical variable
> smk.tab <- table(babies$smokef)
> smk.tab

not now yes now 
    742     484 
> addmargins(smk.tab)

not now yes now     Sum 
    742     484    1226 
> 
```

Comparing Two Categorical Variables

Notice that the function `prop.table()` gives you the overall proportions, i.e., the values in the table add up to 1. For example, 44.7% of the mothers had firstborn babies and were not smokers among all mothers in the dataset: 44.7% = 548/1226 * 100%.



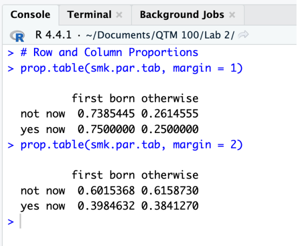
```
> # Summarizing two categorical variables
> smk.par.tab <- table(babies$smokef,babies$parityf)
> addmargins(smk.par.tab)

          first born otherwise  Sum
  not now        548       194  742
  yes now        363       121  484
  Sum            911       315 1226
> prop.table(smk.par.tab)

          first born  otherwise
  not now 0.44698206 0.15823817
  yes now 0.29608483 0.09869494
>
```

```
smk.par.tab <- table(x =
babies$smokef, y =
babies$parityf)
addmargins(smk.par.tab)
prop.table(smk.par.tab)
```

Row and Column Margins



```
> # Row and Column Proportions
> prop.table(smk.par.tab, margin = 1)

         first born otherwise
  not now  0.7385445 0.2614555
  yes now  0.7500000 0.2500000
> prop.table(smk.par.tab, margin = 2)

         first born otherwise
  not now  0.6015368 0.6158730
  yes now  0.3984632 0.3841270
> 
```

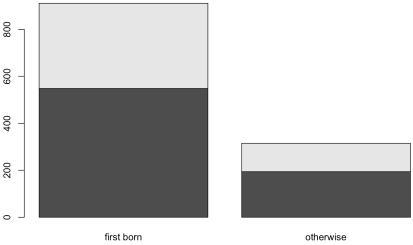
When calculating a row proportion, the denominator is the sum for the row, i.e., values in each row add up to 1 (use `margin = 1`)) . In the case of column proportions, the values in each column add up to 1. (use `margin = 2`))

```
prop.table(smk.par.tab, margin = 1)
prop.table(smk.par.tab, margin = 2)
```

Simple `barplot()`

Basic visualizations for categorical variables include **pie charts** and **bar plots**. When creating these graphs, we need to produce them based on the `table` of the variable(s).

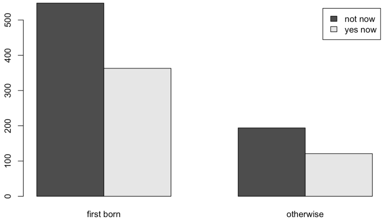
Look at the graph. What does it mean? Is it easy to understand?



`barplot(smk.par.tab)`

Modifying Plots

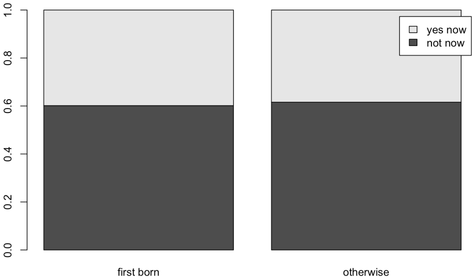
Side-by-side bar plots can be produced using the same function, using the command `beside`. Whenever we use counts, we always use side-by-side bar plots. It is not acceptable to utilize stacked bar plots.



```
barplot(smk.par.tab, beside =
T, legend.text = T)
```

More Modifications

When comparing two groups in a bar chart, it is often best to use proportions in your bar plots instead of counts!



```
barplot(prop.table(smk.par.tab, margin=2), beside = F,
legend.text = T)
```