Introduction to Data in R

##Introduction

```r
library(openintro)
```

```
## Please visit openintro.org for free statistics materials
```

```
##
## Attaching package: 'openintro'
```

```
## The following objects are masked from 'package:datasets':
##
##     cars, trees
```

```r
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:openintro':
##
##     diamonds
```

```r
data(hsb2)
summary(hsb2)
```

```
##        id            gender              race                ses
##  Min.   :  1.00   Length:200         Length:200         low   :47
##  1st Qu.: 50.75   Class :character   Class :character   middle:95
##  Median :100.50   Mode  :character   Mode  :character   high  :58
##  Mean   :100.50
##  3rd Qu.:150.25
##  Max.   :200.00
##      schtyp              prog          read           write            math
##  public :168   general   : 45   Min.   :28.00   Min.   :31.00   Min.   :33.00
##  private: 32   academic  :105   1st Qu.:44.00   1st Qu.:45.75   1st Qu.:45.00
##                vocational: 50   Median :50.00   Median :54.00   Median :52.00
##                                 Mean   :52.23   Mean   :52.77   Mean   :52.65
##                                 3rd Qu.:60.00   3rd Qu.:60.00   3rd Qu.:59.00
##                                 Max.   :76.00   Max.   :67.00   Max.   :75.00
##     science          socst
##  Min.   :26.00   Min.   :26.00
##  1st Qu.:44.00   1st Qu.:46.00
##  Median :53.00   Median :52.00
##  Mean   :51.85   Mean   :52.41
##  3rd Qu.:58.00   3rd Qu.:61.00
##  Max.   :74.00   Max.   :71.00
```

```r
str(hsb2)
```

```
## 'data.frame':    200 obs. of  11 variables:
##  $ id     : int  70 121 86 141 172 113 50 11 84 48 ...
##  $ gender : chr  "male" "female" "male" "male" ...
##  $ race   : chr  "white" "white" "white" "white" ...
##  $ ses    : Factor w/ 3 levels "low","middle",..: 1 2 3 3 2 2 2 2 2 2 ...
##  $ schtyp : Factor w/ 2 levels "public","private": 1 1 1 1 1 1 1 1 1 1 ...
##  $ prog   : Factor w/ 3 levels "general","academic",..: 1 3 1 3 2 2 1 2 1 2 ...
##  $ read   : int  57 68 44 63 47 44 50 34 63 57 ...
##  $ write  : int  52 59 33 44 52 52 59 46 57 55 ...
##  $ math   : int  41 53 54 47 57 51 42 45 54 52 ...
##  $ science: int  47 63 58 53 53 63 53 39 58 50 ...
##  $ socst  : int  57 61 31 56 61 61 61 36 51 51 ...
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
glimpse(hsb2)
```

```
## Observations: 200
## Variables: 11
## $ id      <int> 70, 121, 86, 141, 172, 113, 50, 11, 84, 48, 75, 60, 95, 104...
## $ gender  <chr> "male", "female", "male", "male", "male", "male", "male", "...
## $ race    <chr> "white", "white", "white", "white", "white", "white", "afri...
## $ ses     <fct> low, middle, high, high, middle, middle, middle, middle, mi...
## $ schtyp  <fct> public, public, public, public, public, public, public, pub...
## $ prog    <fct> general, vocational, general, vocational, academic, academi...
## $ read    <int> 57, 68, 44, 63, 47, 44, 50, 34, 63, 57, 60, 57, 73, 54, 45,...
## $ write   <int> 52, 59, 33, 44, 52, 52, 59, 46, 57, 55, 46, 65, 60, 63, 57,...
## $ math    <int> 41, 53, 54, 47, 57, 51, 42, 45, 54, 52, 51, 51, 71, 57, 50,...
## $ science <int> 47, 63, 58, 53, 53, 63, 53, 39, 58, 50, 53, 63, 61, 55, 31,...
## $ socst   <int> 57, 61, 31, 56, 61, 61, 61, 36, 51, 51, 61, 61, 71, 46, 56,...
```

##Types of Variables Categorical data are often stored as factors in R.

Recall from the video that the filter() function from dplyr allows you to filter a dataset to create a subset containing only certain levels of a variable.

Subset of emails with big numbers: email50_big

```
hsb_big <- hsb2 %>%
  filter(gender=="male")
```

Glimpse the subset

```
glimpse(hsb_big)
```

```
## Observations: 91
## Variables: 11
## $ id      <int> 70, 86, 141, 172, 113, 50, 11, 84, 48, 75, 60, 95, 104, 38,...
## $ gender  <chr> "male", "male", "male", "male", "male", "male", "male", "ma...
## $ race    <chr> "white", "white", "white", "white", "white", "african ameri...
## $ ses     <fct> low, high, high, middle, middle, middle, middle, middle, mi...
## $ schtyp  <fct> public, public, public, public, public, public, public, pub...
## $ prog    <fct> general, general, vocational, academic, academic, general, ...
## $ read    <int> 57, 44, 63, 47, 44, 50, 34, 63, 57, 60, 57, 73, 54, 45, 42,...
## $ write   <int> 52, 33, 44, 52, 52, 59, 46, 57, 55, 46, 65, 60, 63, 57, 49,...
## $ math    <int> 41, 54, 47, 57, 51, 42, 45, 54, 52, 51, 51, 71, 57, 50, 43,...
## $ science <int> 47, 58, 53, 53, 63, 53, 39, 58, 50, 53, 63, 61, 55, 31, 50,...
## $ socst   <int> 57, 31, 56, 61, 61, 61, 36, 51, 51, 61, 61, 71, 46, 56, 56,...
```

Table of gender variable

```r
table(hsb_big$gender)
```

```
##
## male
##   91
```

Another table of number variable

```r
table(hsb_big$gender)
```

```
##
## male
##   91
```

#Load Email data

```r
data(email50)
```

Calculate median number of characters: med_num_char

```r
med_num_char <- median(email50$num_char)
```

Create num_char_cat variable in email50

```r
email50 <- email50 %>%
  mutate(num_char_cat = ifelse(num_char < med_num_char, "below median", "at or above median"))
```

Count emails in each category

```r
table(email50$num_char_cat)
```

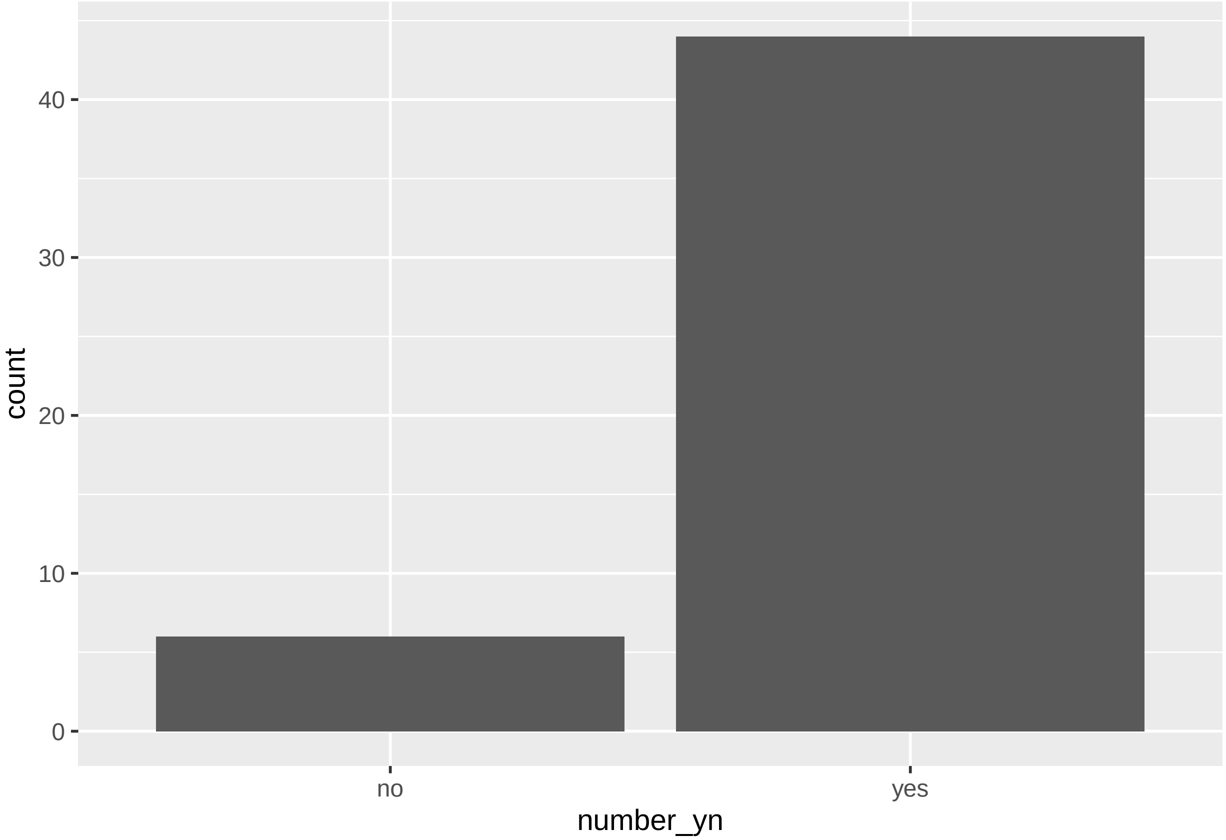
```
##
## at or above median       below median
##                 25                 25
```

Create number_yn column in email50

```
email50 <- email50 %>%
  mutate(number_yn= ifelse(number=="none", "no", "yes"))
```

Visualize number_yn

```
ggplot(email50, aes(x = number_yn)) +
  geom_bar()
```



# Load packages

```
library(tidyr)
```

Count number of male and female applicants admitted

```
hsb_race <- hsb2 %>%
  count(gender, race)
```

View result

```
print(hsb_race)
```

```
## # A tibble: 8 x 3
##   gender race                 n
##   <chr>  <chr>            <int>
## 1 female african american    13
## 2 female asian                8
## 3 female hispanic            11
## 4 female white               77
## 5 male   african american     7
## 6 male   asian                3
## 7 male   hispanic            13
## 8 male   white               68
```

Spread the output across columns

```
hsb_race %>%
  spread(gender, n)
```

```
## # A tibble: 4 x 3
##   race             female  male
##   <chr>             <int> <int>
## 1 african american     13     7
## 2 asian                 8     3
## 3 hispanic             11    13
## 4 white                77    68
```

Table of counts of admission status and gender

count(schtyp, gender) %>% # Spread output across columns based on admission status spread(schtyp, n) %>% # Create new variable mutate(Perc_type = public/ (public+private)) print(hsb_type) # Table of counts of admission status and gender for each department hsb_type2 <- hsb2 %>% count(ses, schtyp, gender) %>% spread(schtyp, n) %>% # Percentage of those admitted to each department mutate(Perc_type = public / (public+private)) print(hsb_type2)

library(openintro) data(county)

county_noDC<- county %>% filter(state !="District of Columbia") %>% droplevels()

#Simple Random Sample
county_srs <- county_noDC %>% sample_n(size=150) glimpse(county_srs)

county_srs %>% group_by(state) %>% count()

#Stratified Sample county_str <- county_noDC %>% group_by(state) %>% sample_n(size=3) glimpse(county_str)

#Beauty in the Classroom download.file("http://www.openintro.org/stat/data/evals.RData", destfile = "evals.RData") load("evals.RData")

Inspect variable types

glimpse(evals) str(evals) # Another option

Remove non-factor variables from this vector

cat_vars <- c("rank", "ethnicity", "gender", "language", "cls_level", "cls_profs", "cls_credits", "pic_outfit", "pic_color")

Recode cls_students as cls_type: evals

evals <- evals %>% # Create new variable mutate(cls_type = factor(ifelse(cls_students <= 18, “small”, ifelse(cls_students >= 19 & cls_students <= 59, “midsize”, “large”))))

Scatterplot of score vs. bty_avg

ggplot(evals, aes(x=bty_avg, y=score)) + geom_point()

Scatterplot of score vs. bty_avg colored by cls_type

ggplot(evals, aes(x=bty_avg, y=score, color=cls_type)) + geom_point()