Module 2: univariate analysis

BIOSTAT 213: Introduction to Computing in the R Software Environment

Yea-Hung Chen, PhD, MS

Wednesday, January 23, 2019

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Working directories and workspaces

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Working directories

- the working directory is the folder on your computer that R will use when opening and saving files
- you can set the working directory according to your preference or whatever you happen to be working on

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Working directories

- on your computer, you probably have different folders for different projects
- for example, I have the following folders on my computer:
  - `module 2`: a folder for Module 2 (today)
  - `module 1`: a folder for Module 1 (last week)
- so, for today's BIOSTAT 213 work, it would make sense for me to set the working directory on my computer to be `module 1`

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Changing the working directory

- if you already have one or more scripts in the desired working directory, an easy way to change the working directory is as follows:
  1. Close RStudio, if it is already open.
  2. Use your computer's file manager (eg, Finder on Apple or File Explorer on Windows) to navigate to the desired working directory.
  3. Open a script (eg, double-click).
     - the script should open in RStudio, if defaults are set
     - instructions for changing defaults on macOS
     - instructions for changing defaults on Windows
- RStudio should automatically set the working directory

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Checking the working directory

- the working directory is always displayed in RStudio's Console pane
- alternatively, use: `getwd()` (short for get working directory)

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Other ways to change the working directory

- using the menu: Session » Set Working Directory » Choose Directory...
- or, use `setwd()` (short for set working directory)
- for example:

```
> setwd('/home/yea-hung/research/project 1')
```

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Workspaces

- R will try to save all of the objects you define
- collectively, those objects are known as the workspace
- every time you quit RStudio or R, you will be prompted to indicate whether or not you want to save the workspace
- you can also manually save the workspace at any time using `save.image()`

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Workspaces

- by default, the workspace is saved to a hidden file (`.RData`) in the working directory
- that file is known as the image
- each working directory you use may have its own image
- returning to the previous example, if I save my workspaces, there will be an image in `module 1` and an image in `module 2`

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Summary

- the working directory is the folder on your computer that R will use when opening and saving files
- each working directory may contain its own image, which is a saved representation of the all objects defined (the workspace) while you were working in the working directory

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Suggestions

- working directories are extraordinarily useful (as we'll see in the next section)
- I tend not to focus, however, on whether or not objects are saved (ie, I save images but don't rely on this as a method for saving work)
- instead, I focus on saving my work by saving my code to scripts

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Brief Exercise 1

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 1

You may work in groups of 2–3. Save your code in a script file.

1. If you have not already, use your computer's file manager (eg, Finder on Apple or File Explorer on Windows) to create a folder on your computer for today's work. Use any arbitrary name for the folder (eg, `module 2`).
2. Use RStudio to create a new script: File » New File » R Script. Alternatively, use the icon or keyboard shortcut (Ctrl + ⇧ + N).
3. Save the file in the folder from (1): File » Save. Alternatively, use the icon or keyboard shortcut (Ctrl + S). Use any arbitrary file name, but make sure to give the file a `.r` or `.R` extension.

More on the next slide...

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Brief Exercise 1

4. In the script, write code for saving your first name.
5. Run the code you wrote above: place the cursor somewhere in the line and press Ctrl + ↵ (alternatively, use the icon or the menu).
6. In the console, type the name of the object you defined above, and press ↵. Verify that your computer returns the value of the object (your first name).

More on the next slide...

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Brief Exercise 1

7. Close RStudio. When prompted to indicate whether to save the workspace image, select Save.
8. Use your computer's file manager to navigate to the folder you created for today's work. Open the script file you created for today's work. Verify that the file opens in RStudio (if not, use the links provided earlier to adjust your computer's settings).
9. In the console, type the name of the object you defined above, and press ↵. Verify that your computer returns the value of the object (your first name).
10. Verify that the correct working directory is displayed in the Console pane. Also verify the working directory using `getwd()`.

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Data frames

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Importing CSV files

- CSV (comma-separated values) file: a plain-text data file, with values separated by commas
- to import a CSV file, use `read.csv()`

```
> ii<-read.csv('nhanes.csv')
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Importing CSV files

```
> ii<-read.csv('nhanes.csv')
```

- in the example above:
  - `read.csv`: the name of the function
  - `nhanes.csv`: the filename, which must be in quotes
  - `<-`: a left arrow, indicating that you want to save the data
  - `ii`: an arbitrary name for the data
- you might imagine, for example, placing (<-) data in a file folder and then labeling the folder (`ii`)

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Importing CSV files

```
> ii<-read.csv('nhanes.csv')
```

- this works because the file is in my working directory
- if the file was in a different directory, I would have to specify the directory in `read.csv()`

```
> ii<-read.csv('~/Downloads/data/nhanes/h/nhanes.csv')
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

The NHANES data file

- `nhanes.csv` contains data from the 2013–2014 implementation of the National Health and Nutrition Examination Survey (NHANES)
- the file contains "original" variables as well as some variables I added
- descriptions of the original variables are available here
- I restricted the data to individuals 20 years of age or older ("adults")

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Data frames

```
> class(ii)
[1] "data.frame"
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

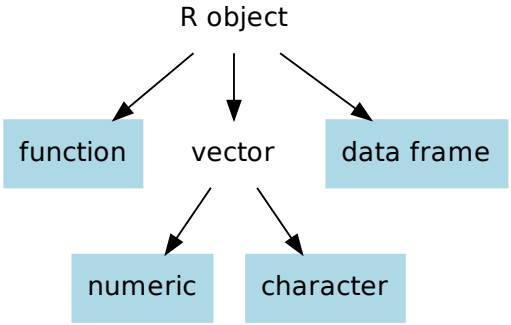
Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Data frames



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Data frames

- data frame: two-dimensional object, analogous to spreadsheets
- data frames have column names
- they can also have row names

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Exploring data frames

- in RStudio, you can view the data by clicking on the object name in the Environment pane

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Exploring data frames

- `names()` for column names
- `head()` for the first few rows
- `tail()` for the last few rows
- `str()` for the first few rows, as well as object types (more on this soon)

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Exploring data frames

- `dim()` for the number of rows and columns
- `nrow()` for the number of rows
- `ncol()` for the number of columns

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Exploring data frames

```
> dim(ii)
[1] 5769  437
> nrow(ii)
[1] 5769
> ncol(ii)
[1] 437
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Accessing variables in data frames

```
> mean(ridageyr)
```

- the above results in an error, even though there is a variable called `ridageyr` in the NHANES data

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Accessing variables in data frames

```
> mean(ii$ridageyr)
[1] 49.11111
```

- in the example above:
  - `mean`: a function for finding the arithmetic mean
  - `ii`: the name of the data frame
  - `$`: indicates that `ridageyr` is a column in `ii`
  - `ridageyr`: the variable name
- I often refer to the pattern above as dollar-sign notation

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Adding variables to data frames

- to add variables to the data frame, use assignment

```
> ii$id<-1:nrow(ii)
```

- in the example above:
  - `1:nrow(ii)`: sequential integers
  - `<-`: a left arrow, indicating we want to save the sequential integers
  - `ii`: the name of the data frame
  - `$`: the dollar sign
  - `id`: the variable name (the variable does not yet exist in `ii`)

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Adding variables to data frames

- a check of the work:

```
> head(ii$id)
[1] 1 2 3 4 5 6
> tail(ii$id)
[1] 5764 5765 5766 5767 5768 5769
```

- this example also shows that `head()` and `tail()` work on vectors (not just on data frames)

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Adding variables to data frames

- if, for some reason, you want to add a single value as a variable to the data frame (with the single value repeating down rows), you can just specify the single value

```
> ii$one<-1
```

- in other words, the following is equivalent, but unnecessary:

```
> ii$one<-rep(1,nrow(ii))
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Replacing variables in data frames

- the previous example demonstrates that it is possible to replace variables in data frames using assignment

```
> head(ii$ridageyr)
[1] 69 54 72 73 56 61
> ii$ridageyr<-45
> head(ii$ridageyr)
[1] 45 45 45 45 45 45
```

- notice that R does not provide any warning

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

More on variables in data frames

```
> head(ii$bmxbmi)
[1] 26.7 28.6 28.9 19.7 41.7 35.7
> class(ii$bmxbmi)
[1] "numeric"
```

- each variable in each data frame has a class

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

More on variables in data frames

```
> head(ii$ridageyr)
[1] 69 54 72 73 56 61
> class(ii$ridageyr)
[1] "integer"
> is.numeric(ii$ridageyr)
[1] TRUE
```

- minor note: I forgot to mention last week that there is a "sub-class" of numeric objects known as integer objects

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Integer objects

```
> class(1:10)
[1] "integer"
> is.numeric(1:10)
[1] TRUE
> is.vector(1:10)
[1] TRUE
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

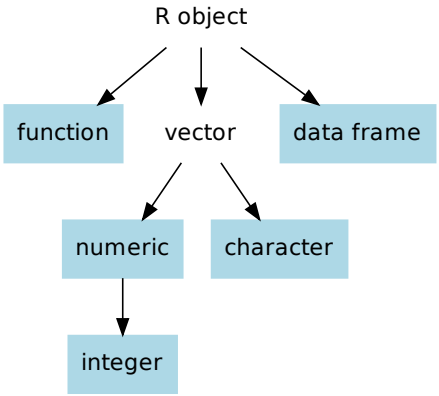
Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Integer objects



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Creating data frames

```
> students<-data.frame(first=c('Bob','Mary','Jane'),
+                      hw1=c(86,92,95),hw2=c(92,84,93))
> students
  first hw1 hw2
1   Bob  86  92
2  Mary  92  84
3  Jane  95  93
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 2

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 2

You may work in groups of 2–3. Save your code in a script file.

1. Download the NHANES data file, `nhanes.csv` from the Collaborative Learning Environment. Depending on your operating system or browser, you may need to right-click. Save the file to the folder you created for today's work.
2. Import the data into R.
3. Explore the data frame using RStudio's built-in data viewer.
4. Explore the data frame using `names()`, `head()`, `tail()`, and `dim()`.

More on the next slide...

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Brief Exercise 2

5. The variable `bmxwt` stores weight in kilograms. Use this variable to add a new variable to the data frame for weight in pounds (1 kg = 2.20462 lb).
6. The variable `bmxht` stores height in centimeters. Use this variable to add a new variable to the data frame for height in feet (1 cm = 0.0328084 ft).

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Summarizing continuous data

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Mean, standard deviation, and variance

```
> mean(ii$ridageyr)
[1] 49.11111
> sd(ii$ridageyr)
[1] 17.5632
> var(ii$ridageyr)
[1] 308.466
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Mean, standard deviation, and variance

```
> mean(ii$bmxbmi)
[1] NA
> mean(ii$bmxbmi,na.rm=TRUE)
[1] 29.10275
> sd(ii$bmxbmi)
[1] NA
> sd(ii$bmxbmi,na.rm=TRUE)
[1] 7.153478
```

- if there are missing values, use the `na.rm=TRUE` argument
- more on missing values in Module 4

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Mean, standard deviation, and variance

```
> var(ii$ridageyr)
[1] 308.466
> sum((ii$ridageyr-mean(ii$ridageyr))^2)/(length(ii$ridageyr)-1)
[1] 308.466
```

- `sd()` and `var()` use a denominator of $n - 1$ (more on this in your statistics classes)

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Quantiles

```
> median(ii$ridageyr)
[1] 48
> quantile(ii$ridageyr,0.5)
50%
 48
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Quantiles

```
> quantile(ii$ridageyr,c(0.25,0.5,0.75))
25% 50% 75%
 34  48  63
```

- the c() is necessary because the second argument should be a numeric vector
- in other words, the above is short for:

```
> quantile(ii$ridageyr,probs=c(0.25,0.5,0.75))
25% 50% 75%
 34  48  63
```

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Minimums and maximums

```
> min(ii$ridageyr)
[1] 20
> max(ii$ridageyr)
[1] 80
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Summaries

```
> summary(ii$ridageyr)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  20.00   34.00   48.00   49.11   63.00   80.00
```

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Histograms

```
> hist(ii$ridageyr)
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

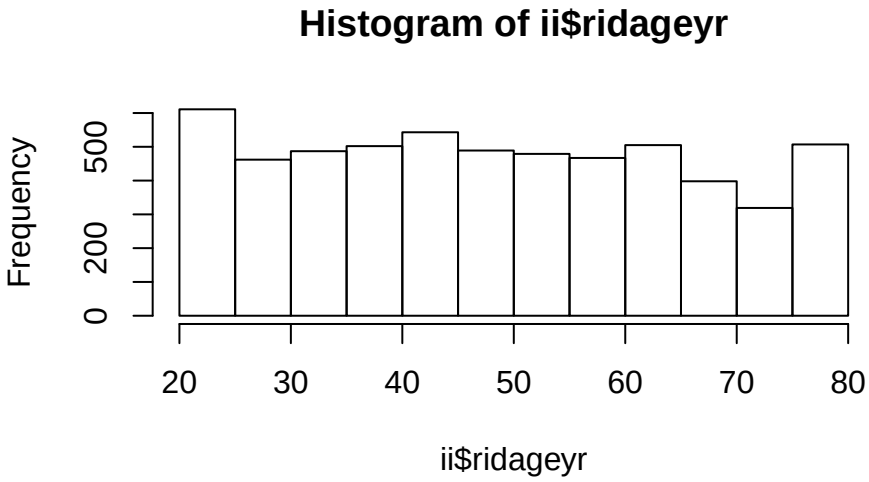
Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Histograms



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Histograms

```
> hist(ii$ridageyr,main='',xlab='Age')
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

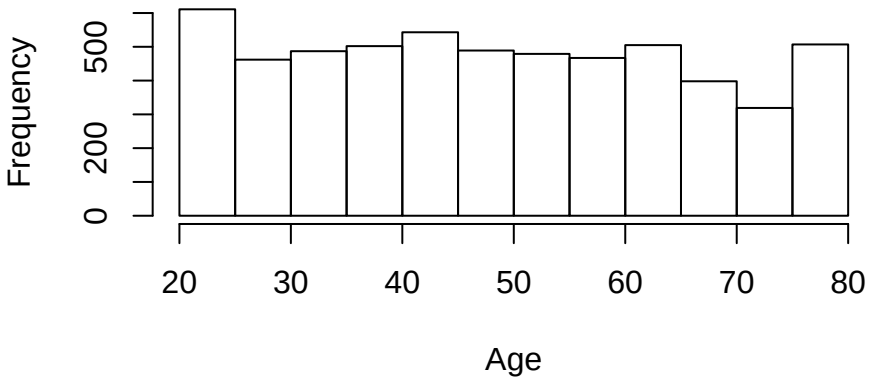
Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Histograms



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Histograms

```
> hist(ii$ridageyr,breaks=seq(20,80,10),
+      main='',xlab='Age')
```

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

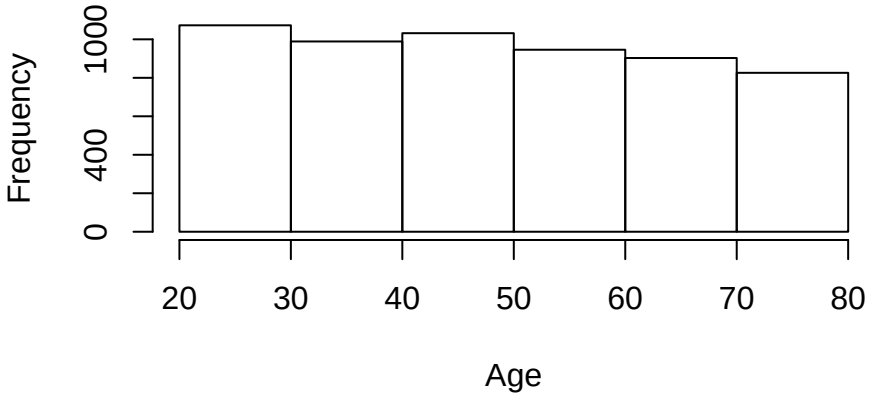
Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Histograms



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Histograms

```
> hist(ii$ridageyr,freq=FALSE,main='',xlab='Age')
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

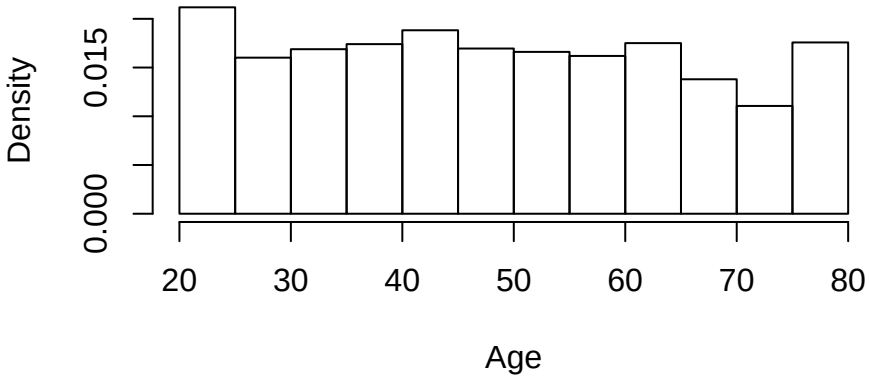
Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Histograms



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Histograms

- we'll discuss an alternate method for creating histograms next week

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 3

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 3

You may work in groups of 2–3. Save your code in a script file.

1. Find the mean and standard deviation of levels of high-density lipoprotein (`lbdhdd`).
2. Find the 20th, 40th, 60th, and 80th percentiles of levels of high-density lipoprotein (`lbdhdd`).
3. Produce a histogram of levels of high-density lipoprotein (`lbdhdd`).

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Factors

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Factors

```
> head(ii$gender)
[1] Male   Male   Male   Female Male   Female
Levels: Female Male
> class(ii$gender)
[1] "factor"
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Factors

- a factor is an object that has a limited set of possible values
- the possible values are known as levels
- the levels have order (and it is possible to change the order of the levels)
- these orders can be useful for plots and for regression models
- factors are useful for storing categorical variables

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Factors

```
> head(ii$gender)
[1] Male   Male   Male   Female Male   Female
Levels: Female Male
> class(ii$gender)
[1] "factor"
> levels(ii$gender)
[1] "Female" "Male"
```

- gender has 2 levels: Female and Male
- the levels are ordered: the first level is Female

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

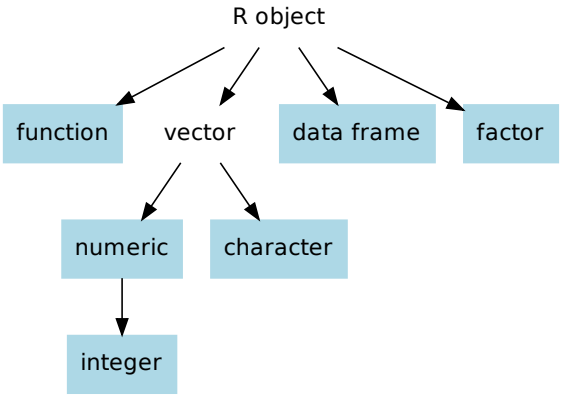
Factors

```
> length(ii$gender)
[1] 5769
> is.vector(ii$gender)
[1] FALSE
```

- minor note: factors look like vectors, but are not vectors

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Factors



Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Reordering levels

```
> levels(ii$gender)
[1] "Female" "Male"
> ii$gender<-relevel(ii$gender,'Male')
> levels(ii$gender)
[1] "Male"   "Female"
```

- the second argument is the level that we want to be first
- all other levels are shifted over

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Reordering levels

- for greater control, use `factor()`

```
> levels(ii$bpcat)
[1] "Elevated"            "Hypertensive crisis"
[3] "Normal"              "Stage 1"
[5] "Stage 2"
> ii$bpcat<-factor(ii$bpcat,c('Normal','Elevated','Stage 1',
+                             'Stage 2','Hypertensive crisis'))
> levels(ii$bpcat)
[1] "Normal"              "Elevated"
[3] "Stage 1"             "Stage 2"
[5] "Hypertensive crisis"
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Renaming levels

```
> levels(ii$hs)
[1] "No"      "Smoking"
> levels(ii$hs)<-c('No','History of smoking')
> levels(ii$hs)
[1] "No"                 "History of smoking"
```

- this sort of renaming should be done cautiously (more on data cleaning and preparation in Module 4)

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Factor versus character

- factor objects are useful for categorical variables (eg, US state)
- character objects are more useful for free response (eg, street address)

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

`read.csv()` and factors

- by default, if a variable in the data file has a character string in it, `read.csv()` will treat the variable as a factor (to override that default, use the `stringsAsFactors=FALSE` argument)
- the default ordering of the levels is alphabetical

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Factor to character, and vice versa

```
> levels(ii$bpcat)
[1] "Normal"                "Elevated"
[3] "Stage 1"               "Stage 2"
[5] "Hypertensive crisis"
> ii$bpcat<-as.character(ii$bpcat)
> class(ii$bpcat)
[1] "character"
> ii$bpcat<-as.factor(ii$bpcat)
> class(ii$bpcat)
[1] "factor"
> levels(ii$bpcat)
[1] "Elevated"              "Hypertensive crisis"
[3] "Normal"                "Stage 1"
[5] "Stage 2"
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Factors

- more on factors and data preparation in Module 4

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Counts and percents

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Counts

```
> table(ii$bpcat)

            Normal           Elevated            Stage 1
              2307                829               1036
          Stage 2 Hypertensive crisis
               886                 53
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Counts

```
> table(ii$bpcat,useNA='always')

            Normal           Elevated            Stage 1
              2307                829               1036
           Stage 2 Hypertensive crisis               <NA>
               886                 53                658
```

- use the `useNA='always'` argument to show missing values

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Counts

- `table()` works with numeric and character objects, not just factors

```
> class(ii$sleep)
[1] "integer"
> table(ii$sleep)

   2    3    4    5    6    7    8    9   10   11   12
  22   38  230  523 1381 1518 1552  304  144    9   39
```

Module 2: univariate analysis
Yea-Hung Chen, PhD, MS
Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Percents

```
> prop.table(table(ii$bpcat))

            Normal           Elevated            Stage 1
        0.45137938         0.16219918         0.20270006
           Stage 2 Hypertensive crisis
        0.17335159         0.01036979
```

- you must include `table()` inside `prop.table()` (it is easy to forget!)

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Percents

```
> prop.table(table(ii$bpcat))*100

           Normal          Elevated           Stage 1
        45.137938         16.219918         20.270006
          Stage 2 Hypertensive crisis
        17.335159          1.036979
```

Module 2: univariate analysis

Yea-Hung Chen, PhD, MS

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 4

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 4

You may work in groups of 2–3. Save your code in a script file.

1. Reorder the levels of low-density lipoprotein (`ldlcat`), using the following order: `Optimal`, `Near optimal`, `Borderline high`, `High`, and `Very high`.
2. Find the number of individuals in each `ldlcat` group.
3. Find the percent of individuals in each `ldlcat` group.

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate hypothesis tests

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Univariate *t*-tests

```
> t.test(ii$bmxbmi,mu=23)
```

- use the `mu=` argument to specify the null hypothesis
- in other words, this tests the null hypothesis that the mean body mass index (BMI) among American adults is 23 kg/m^2

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate *t*-tests

```
> t.test(ii$bmxbmi,mu=23)

    One Sample t-test

data:  ii$bmxbmi
t = 63.384, df = 5519, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 23
95 percent confidence interval:
 28.91400 29.29151
sample estimates:
mean of x
 29.10275
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate *t*-tests

```
> tt<-t.test(ii$bmxbmi,mu=23)
> names(tt)
[1] "statistic"   "parameter"   "p.value"     "conf.int"
[5] "estimate"    "null.value"  "alternative" "method"
[9] "data.name"
```

- it is often useful to save hypothesis tests
- the `names()` function provides an easy way to see accessible parts of the stored object

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate *t*-tests

```
> names(tt)
[1] "statistic"   "parameter"   "p.value"     "conf.int"
[5] "estimate"    "null.value"  "alternative" "method"
[9] "data.name"
> tt$statistic
       t
63.38378
> tt$parameter
  df
5519
> tt$p.value
[1] 0
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate *t*-tests

```
> tt$estimate
mean of x
 29.10275
> tt$conf.int
[1] 28.91400 29.29151
attr(,"conf.level")
[1] 0.95
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Univariate χ^2-tests

```
> prop.table(table(ii$hs))

                No History of smoking
           0.561297            0.438703
```


Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Univariate χ^2-tests

```
> chisq.test(table(ii$hs),p=c(0.5,0.5))
```

- use the p= argument to specify the null hypothesis
- this tests the null hypothesis that 50% of American adults have a history of smoking

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate χ^2-tests

```
> chisq.test(table(ii$hs),p=c(0.5,0.5))

    Chi-squared test for given probabilities

data:  table(ii$hs)
X-squared = 86.674, df = 1, p-value < 2.2e-16
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Univariate χ^2-tests

```
> tt<-chisq.test(table(ii$hs),p=c(0.5,0.5))
> names(tt)
[1] "statistic" "parameter" "p.value"   "method"
[5] "data.name" "observed"  "expected"  "residuals"
[9] "stdres"
> tt$statistic
X-squared
 86.67401
> tt$parameter
df
 1
> tt$p.value
[1] 1.279677e-20
```


Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5

Brief Exercise 5

Working directories and workspaces

Brief Exercise 1

Data frames

Brief Exercise 2

Summarizing continuous data

Brief Exercise 3

Factors

Counts and percents

Brief Exercise 4

Univariate hypothesis tests

Brief Exercise 5

Brief Exercise 5

You may work in groups of 2–3. Save your code in a script file.

1. Find the mean level of low-density lipoprotein (`lbdldl`) in the sample.
2. Use the `lbdldl` variable to conduct a univariate t-test of the null hypothesis that the mean level of low-density lipoprotein among American adults is 100 mg/dL (a threshold for "near optimal").

Working directories and workspaces
Brief Exercise 1
Data frames
Brief Exercise 2
Summarizing continuous data
Brief Exercise 3
Factors
Counts and percents
Brief Exercise 4
Univariate hypothesis tests
Brief Exercise 5