Week 2 2.2 Discover Note

Lecture 4: Data and Qualitative Data

1. Australian Road Fatality Data

· Domain Knowledge

Despite preventative measures such as *compulsory seat belts (from 1970)* and *school zones (2001)*, the number of road fatalities in Australia **continues to rise**.

- In 2015, 1,209 died in road fatalities, why?
- We are going to investigate data from the *Australian Bureau of Statistics (ABS)* for Jan-Apr 2016.

> * Data: **Information** about the set of **subjects** being studied (ex. road fatalities).

- Most commonly, data refers to the **sample** (parts of the population), not the (entire) **population**.
- Different types of data: survey data, spreadsheet type data, MRI image data, etc.

> * Initial Data Analysis (IDA): **A first general look** at the data without formally answering the research questions.

- It can/may:
  - Help you to see whether the data can **answer your questions**.
  - Pose **other** research **questions**.
  - Identify the **mean qualities** of the data.
  - **Suggest the population** from which a sample derives.
- What's involved?
  - Data Background: Check the **quality and integrity** of the data
  - Data Structure: What **information** has been **collected**?
  - Data **Wrangling** (getting data **ready for analysis**): Scarping, cleaning, tidying, reshaping, splitting, combining
  - Data **Summaries**: Graphical and numerical
- For **qualitative and quantitative** data, we focus on **structure & graphical** summaries.
- Steps:
  - List 6 top rows of the dataset: `head(dataset)`
  - List x top row(s) of the dataset: `head(dataset,x)`
  - List the number of rows and number of columns: `dim(dataset)` - `dim` as dimension
  - List the names of the variables: `name(dataset)`
  - Classify variables: `str(dataset)` - `str` as structure
  - Isolate a variable: `data$variable`
  - See the length of a variable: `length(variable)`
  - Calculate the sum of a variable (quantitative): `sum(variable)`
  - Sort data in increasing order: `sort(variable)`
  - Sort data in increasing order: `sort(variable, decreasing=T)`

· Structure of the Data

> * Variable: Something that measures or describes some **attribute** of the subjects.

- Types of Variables:
  - **Qualitative/Categorical (Categories)** (named as **factor** in R)
    - Ordinal (ordered)
      - Binary (2 categories)
      - 3 + categories
    - Nominal (non-ordered)
      - Binary (2 categories)
      - 3 + categories
  - **Quantitative/Numerical (Measurements)** (named as **num** or **numeric** in R)
    - Discrete (separated)
    - Continuous (continuum)
- Data with *p* variables is said to have **dimension** *p*.
  - Univariate: 1 variable
  - Bivariate: 2 variables
  - Multivariate: 2+ variables

> # `str(data)`, *Structure of Data*

· Graphical Summaries

- The aim of a GS is to best **highlight features** of this data.
  - Pie chart is popular but not informative.
  - Try *Shiny Apps* (made by R Studio), present data in accessible ways
- Qualitative Data

> # Bar plot (bar chart, bar graph): is a simple **summary** of qualitative data.
>
> 1. `Dayweek = data$Dayweek` *(select Dayweek variable from the whole data frame by `data$`)*
> 2. `table(Dayweek)` *(Produce a **frequency table** of fatalities per day of the week--**numerical**)*
> 3. `barplot(table(Dayweek))` *(Produce a **bar chart**--**graphical**)* or `plot(table(Dayweek))` *(A simpler version)*
> 4. Find out that fatalities are more common in Saturday.
> 5. Statistical Thinking (15:33~16:10 in lec 4)
>
> ### # Double bar plot: 2 quantitative variables
>
> 1. *Select* `Dayweek` *and* `Gender` *as variables:*

```
Dayweek = data$Dayweek
Gender = data$Gender
```

> 2. *Produce a **double frequency table** (contingency table):*

```
data1 = table(Gender,Dayweek)
data1
```

> 3. *Use* `barplot(data1)` *to create a double bar chart (stacked or side-by-side)*

- Big Data

> * **Big Data: The massive amounts of data being collected in fields.**

- Big data is **high dimensional**, as they are **more variables *p* than subjects *n***.
  - ex. Genomics data can have 3000000 variables.
- Big data requires more complex visualizations.

2. Summary

1 Qual	2 Qual
Simple Bar Plot	Double Bar Plot

1 Quant	2 Quant
Histogram/Box Plot	Scatter Plot

1 Qual + 1 Quant
Comparative (side-by-side) Box Plot

Lecture 5: Quantitative Data

---

1. Australian Road Fatality Data (Continue)

· Primitive Cleaning of the Data (Data Wrangling)

> # `C` is the **cleaned** version of the data: `data = read.csv("data/2016FatalitiesC.scv",header=T")`

2. Histogram

> * Histogram **highlights the percentage of data** in one class interval compared to another.

- It consists of a set of blocks representing the percentages by **area**.
- The area of the whole histogram is **100%**.
- The **horizontal** scale is divided into **class intervals**.
- The **area** of each block represents the **percentage of subjects** in that class interval.
- The **height** of each block measures **crowding**.
- Choices:
  - No need to add vertical scale.
  - We will mostly use the **density scale**:

    > * Density Scale: *Height* of Each Block = *%* in the block / *Length* of the Class Interval

  - For continuous data, we need an **endpoint convention** for data points that **fall on the border** of 2 class intervals.

- If an interval contains the left endpoint but excludes the right endpoint, then an 18 year old would be counted in [18,21), not [0, 18).
- In this case, 18 is the left endpoint, 21 is the right.
- This is called "**left-closed and right-open ([ ))**"

· How to produce a histogram by Hand:

- **1. Construct the distribution table.**

Class Intervals	Number of Subjects in Interval	%	Height of Block
[0,18)	29	6.6	0.004
[18,25)	72	16.4	0.023
[25,70)	259	58.9	0.013
[70,100)	80	18.2	0.006
	440	100	

(Where Height of Block = % per year)

- **2. Draw the horizontal axis and blocks.**

· The Speedy Way in R:

1. Read in data. (line 1)
2. Choose a variable. (line 2 - `Age` is the variable, select it by using `data%`)
3. Choose the class intervals. (line 3 - each number represents an endpoint of interval (0-18, 18-25, etc.) - `breaks=c(0,18,25,70,100)`)
4. Produce a distribution table. (line 4 - It means *"create a **table** of **age cut** by **breaks** that's **right-open**"* - `table(cut(Age,breaks,right=F))`, `F` is False)

or:

4. Produce a histogram. (line 5, 6, 7 - `hist` is histogram, `br` means breaks, `freq=F` means produce the hist on the **density scale**, `xlab` is the name for horizontal/x scale, `ylab` is the name for vertical/y/density scale, and `main` is the title of the hist)

```
data = read.csv("data/2016FatalitiesC.csv,header=T)
Age = data%Age
breaks=c(0,18,25,70,100)
table(cut(Age,breaks,right=F))
or:
hist(Age,br=breaks,freq=F,right=F,
xlab="Age (in years)", ylab="% per year",
main=Histogram for Age of Road Fatality in Australia: Jan-June 2016")
```

· Control for a Variable

1. Select female ages only (line 1), and male ages only (line 2)
2. Put the graphic output in `1` *row* with `2` *columns*
3. Produce a histogram of female ages (line 4) and male ages (line 5) both with density scale

```
AgeF = data$Age[data$Gender=="Female"]
AgeM = data$Age[data$Gender=="Male"]
par(mfrow=c(1,2))
hist(AgeF,freq = F)
hist(AgeM,freq = F)
```

· Common Mistakes of Histogram

1. Make the block heights **equal** to the percentages.
2. Use **too many** class intervals. (10-15 for maximum)

3. Box Plot

- Simple Box Plot
  - Useful for **comparing** multiple **datasets**.
  - It plots the **median**, the middle 50% of the data in a box, and determines any outliers.

```
Age = data$Age
summary(Age)
par(mfrow=c(1,2))
boxplot(Age)
# to make the boxplot horizontally layout, add `horizontal=T`:
boxplot(Age,horizontal=T)
```

- Comparative Box Plots
  - **Split up** a quantitative variable by a quantitative variable.

```
Gender = data$Gender
summary(Age[Gender=="Female"])
summary(Age[Gender=="Male"])
boxplot(Age~Gender,horizontal=T)
```

4. Scattered Plot

- Examine the **relationship** between 2 quantitative variables.

```
Speedlimit = data$Speedlimit
plot(Age,speedlimit)
```

Lecture 6: Data Visualization

---

1. Price Point for a Diamond

· Domain Knowledge

- Diamonds are the hardest known natural material and one of the world's major natural resources.
    - Australia has the largest reserves estimated at around 210 million carats.
- Diamonds are used for jewelry (30%, sales of US79 Billion in 2015) and industrial applications (70%).
    - The 59.6 carat Pink Diamond is currently the most expensive gemstone selling for $71.2 million in 2017.
- Pricing Diamonds: buyers need to investigate the price point as each diamond is unique.
    - Diamonds are graded by 4 qualities, known as the "4Cs":
        - Carat (weight)
        - Cut (quality of cut according to proportions, symmetry and polish)
        - Color (color-graded from D-colorless to Z-saturated)
        - Clarity (graded from flawless to inclusions)

· Diamonds Dataset

- Include the prices and 9 other attributes of 54,000 diamonds.

```
# Diamonds dataset is already in ggplot2:
install.packages("ggplot2")
library(ggplot2)
or:
install.packages("tidyverse")
library(tidyverse)
diamonds
# Look into its data structure:
str(diamonds)
```

· Data Visualization (Data Viz)

- Graphical summaries on steriods.
- What determines a good data visualization?
    - Tell an interesting story in visual appealing way (require understanding on both data and design)
    - ex. What questions are we trying to answer?
    - What variables are we highlighting? ...
- What about poor?
    - Tell a story in a visually boring or distracting way (chartjunk)
- Bad?
    - Tell a misleading story *especially in an appealing way!)

· ggplot2 (Grammar of Graphics Plot 2)

- ggplot allows you to **specify the individual building blocks** of your plot, and then combine them to create just about any kind of visualization you want.
    - The *aesthetic* `aes` is what you can **see**. ex. position, outside color, inside color (fill), shape of points, etc.
    - The *geometric objects* `geom_xxx` are the **actual marks** we put on a plot.

    - The *facet* is a **subset** of the data.
- Steps to use:

    1. Install the package.

```
install.packages("ggplot2")
library(ggplot2)
data = diamonds
```

    2. Check if the data is *tidy*.

- the data needs to be in a **data frame**, with subjects as rows and variables as columns.

> # `head(data,2)`
>
>   3. Check classification of variables.
>   # `str(data)`
>   4. **Sketch** by hand
>
> - Sketch what you want to produce, labelling the variables.
> - Write the code.
>
>   5. Run you ggplot and customise to improve visual design.

- Simple Barplot (1 qual)

```
# Define x axis by 1 variable (cut):
p = ggplot(diamonds,aes(x=cut))
# Represent the data by bar chart:
p + geom_bar()
# Fill the chart with color by a 2nd variable (clarity):
p + geom_bar(aes(fill=clarity))
# Produce a side-by-side graph:
p + geom_bar(aes(fill=clarity),position="dodge")
```

- Histogram (1 quant)

```
p1 = ggplot(diamonds,aes(x=price))
p1 + geom_histogram(aes(fill=cut))
# Change the width of the bar (bin)
p1 + geom_histogram(aes(fill=cut),binwidth=1000)
```

- Simple Scattered Plot (2 quant)

```
# Define x axis as carat, y axis as price:
p2 = ggplot(diamonds,aes(x=carat,y=price))
# Represent data by points:
p2 + geom_point()
# Add 1 quant aesthetic: color by 3rd varaible (carat)
p2 + geom_point(aes(color=carat))
# Add 2 qual aesthetic: color by 3rd varaible (clarity), and 4th variable
(shape):
p2 + geom_point(aes(color=clarity,shape=cut))
```