POLI 30 D: Political Inquiry

TA Sessions

Lab 06 | R Plots and R Data Analysis II

Before we start

Announcements:

- GitHub page: **https://github.com/umbertomig/POLI30Dpublic**
- Piazza forum: The link in the slides needs to be fixed. Check with instructors for an alternative link.

Before we start

Recap: In the Lab sessions, you learned:

- How to install R and R Studio on your computer.
- How to do basic math operations in R.
- How to do basic vector and data.frame operations in R.
- How to install packages and work with R Markdown.
- How to work with advanced R objects and create histograms.

Great job!

- Do you have any questions about these contents?

Plan for Lab 05

- Barplots
- Violinplots
- Scatterplots
- Correlation
- Bivariate Regression

Getting started

Getting started

- To get started, we need to load the datasets we will need in the lab.
- We also need to load the `tidyverse` package, which has all the R functions we use.
- Let's do it, then!

Getting started - tidyverse

- Loading the `tidyverse` library:

```
library(tidyverse)
```

Getting started - Education expenditure data

```
educexp <- read.csv("https://raw.githubusercontent.com/umbe
head(educexp)
```

```
##   education income young urban states
## 1       189   2824 350.7   508     ME
## 2       169   3259 345.9   564     NH
## 3       230   3072 348.5   322     VT
## 4       168   3835 335.3   846     MA
## 5       180   3549 327.1   871     RI
## 6       193   4256 341.0   774     CT
```

Getting started - Chile survey data

```
chilesurv <- read.csv("https://raw.githubusercontent.com/um
head(chilesurv)
```

```
##   statusquo vote voteYES
## 1   3.02460    Y       1
## 2  -3.88851    N       0
## 3   3.69216    Y       1
## 4  -3.09489    N       0
## 5  -3.31488    N       0
## 6  -3.14055    N       0
```

Getting started - Voting

```
voting <- read.csv("https://raw.githubusercontent.com/umber
head(voting)
```

```
##   birth message voted
## 1  1981      no     0
## 2  1959      no     1
## 3  1956      no     1
## 4  1939     yes     1
## 5  1968      no     0
## 6  1967      no     0
```

Intro to plots (revisited)

Intro to plots

- For plots, we will use a package called `ggplot2`.
- Here is a good cheat sheet. This is a great thing to print and has close by when creating plots.
- `ggplot2` is based on the `grammar of graphs`. But what is this?

Intro to plots

- In the abstract, the grammar of graphs is a decomposition of plots in its main features.
- In essence, every plot has the following:

1. A dataset
2. A coordinated system
3. A geometric shape

- And different plots are different compositions of these three key ingredients.

Barplots

Barplots

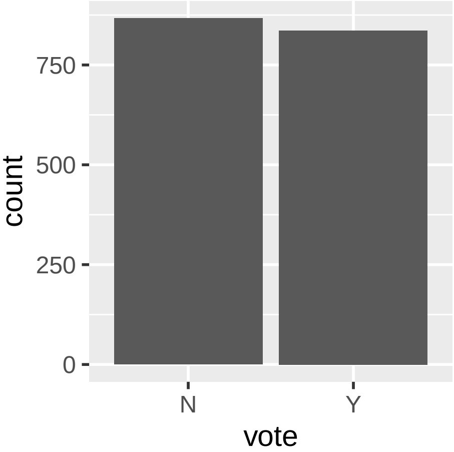
- Barplots is great for representing a binary variable. The basic syntax is:

```
ggplot(data = dataset,
       mapping = aes(x = variable_x_name)) +
  geom_bar()
```

- You need to add `dataset` and the `variable_x_name`.

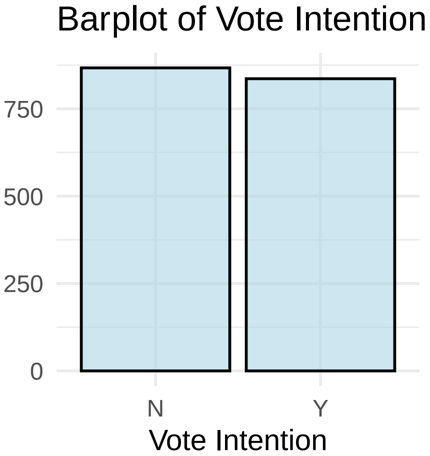
Barplots

```
ggplot(data = chilesurv, aes(x = vote)) +
  geom_bar()
```



Barplots

```
ggplot(data = chilesurv, aes(x = vote)) +
  geom_bar(color = 'black', fill = 'lightblue', alpha = 0.6) +
  labs(title = 'Barplot of Vote Intention', x = 'Vote Intention', y = '') + theme_minimal()
```



Plots for two variables

Plots for two variables

- Most fun things are when we plot one variable against another.
- This is because exploring one variable may be fun, but it could be more informative.
- We want to find relationships between variables!

Plots for two variables

- For this case, whenever a variable is binary or non-binary, we have three combinations with respective plots:

1. Binary x Binary → Mosaic Plots
2. Binary x Non-binary → Violin Plots
3. Non-binary x non-binary → Scatter Plots

Mosaic Plots

Mosaic Plots

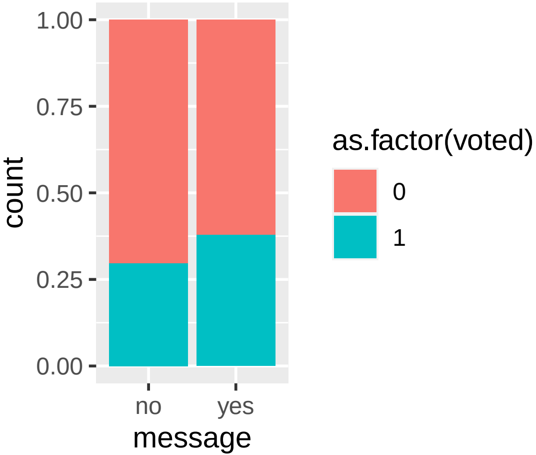
- Type of barplot for two non-binary variables. The syntax is:

```
ggplot(data = dataset,
       mapping = aes(x = treatment_var,
                     fill = as.factor(outc_var))) +
  geom_bar(position = 'fill')
```

- And you need to change the `dataset`, the `outc_var`, and the `treatment_var`.
- The mosaic plots make the relationship between a binary treatment and a binary control very clear!

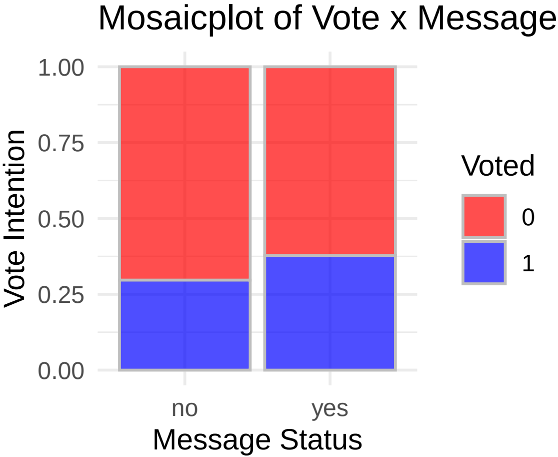
Mosaic Plots

```
ggplot(data = voting, aes(x = message, fill = as.factor(voted))) +
  geom_bar(position = 'fill')
```



Mosaic Plots

```r
ggplot(data = voting, aes(x = message, fill = as.factor(voted))) +
  geom_bar(position = 'fill', alpha = 0.7, color = 'gray') +
  scale_fill_manual(values = c('red', 'blue'), name = 'Voted') +
  labs(title = 'Mosaicplot of Vote x Message',
       x = 'Message Status', y = 'Vote Intention') + theme_minimal()
```



Violin Plots

Violin Plots

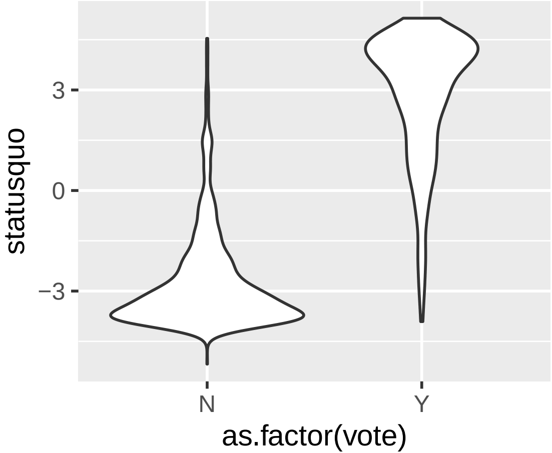
- Violin plot is excellent when you want to check how a non-binary variable and a binary variable are related. The basic syntax is:

```
ggplot(data = dataset,
       mapping = aes(x = as.factor(binary_var),
                     y = nonbin_var)) +
  geom_violin()
```

- And you need to change the `dataset`, the `binary_var`, and the `nonbin_var`.
- But they tend to look somewhat ugly...

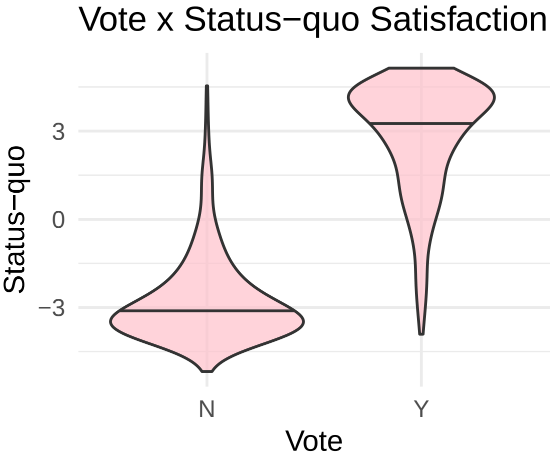
Violin Plots

```
ggplot(data = chilesurv, aes(x = as.factor(vote), y = statusquo)) +
  geom_violin()
```



Violin Plots

```
ggplot(data = chilesurv, aes(x = as.factor(vote), y = statusquo)) +
  geom_violin(fill = 'pink', alpha = 0.7, bw = 0.6, draw_quantiles = 0.5) +
  labs(title = 'Vote x Status-quo Satisfaction',
       x = 'Vote', y = 'Status-quo') + theme_minimal()
```



Scatter Plots

Scatter plots

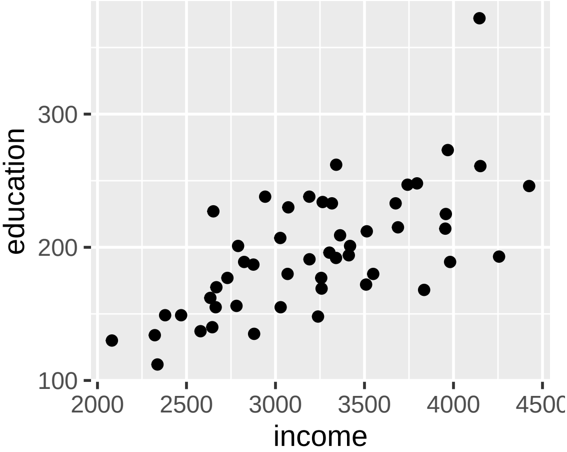
- Scatter plots are great for two non-binary variables. The basic syntax is:

```
ggplot(data = dataset,
       mapping = aes(x = indep_var, y = dep_var)) +
  geom_point()
```

- And you need to change the `dataset`, the `indep_var`, and the `dep_var`.
- They make the relationship between two non-binary variables very clear!
- And you can add a trend line.

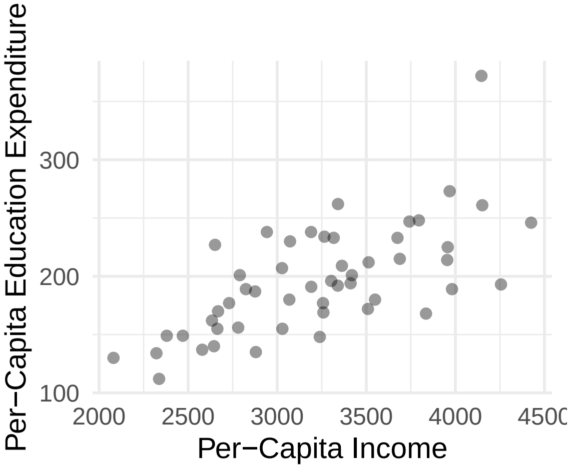
Scatter plots

```
ggplot(data = educexp, aes(x = income, y = education)) +
  geom_point()
```



Scatter plots

```
ggplot(data = educexp, aes(x = income, y = education)) +
  geom_point(fill = 'lightblue', alpha = 0.4) +
  #geom_text(aes(label=states), size=2) +
  labs(title = '', y = 'Per-Capita Education Expenditure', x = 'Per-Capita Income') +
  theme_minimal()
```



Scatter plots with trend line

- To add a trend line, you do the following:

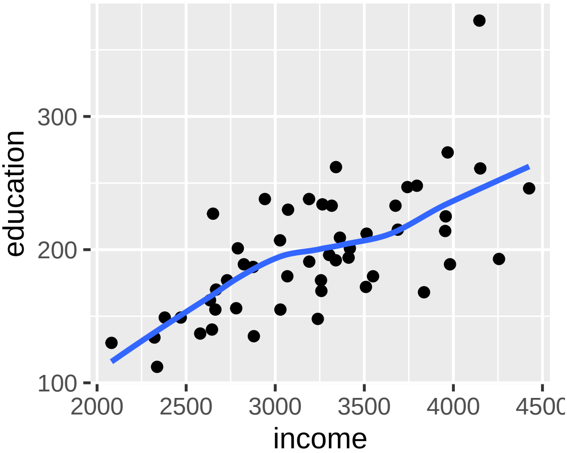
```
ggplot(data = dataset,
       mapping = aes(x = indep_var, y = dep_var)) +
  geom_point() +
  geom_smooth(se = F, formula = 'y ~ x')
```

- And you need to change the `dataset`, the `indep_var`, and the `dep_var`.
- It adds a non-linear trend line called `loess`. To change that, add the `method = 'lm'` parameter!

Scatter plots with trend line

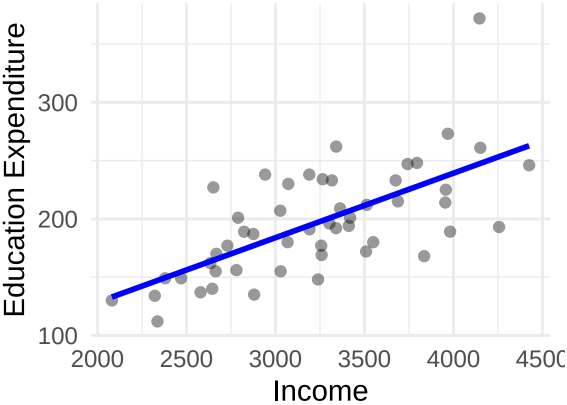
```
ggplot(data = educexp, aes(x = income, y = education)) +
  geom_point() + geom_smooth(se = F, formula = 'y ~ x')
```

```
## `geom_smooth()` using method = 'loess'
```



Scatter plots with trend line

```
ggplot(data = educexp, aes(x = income, y = education)) +
  geom_point(fill = 'lightblue', alpha = 0.4) +
  labs(title = '', y = 'Education Expenditure', x = 'Income') +
  geom_smooth(formula = 'y ~ x', method = 'lm',
              se = F, color = 'blue', lwd = 1) + theme_minimal()
```



Scatter plots with binary dependent

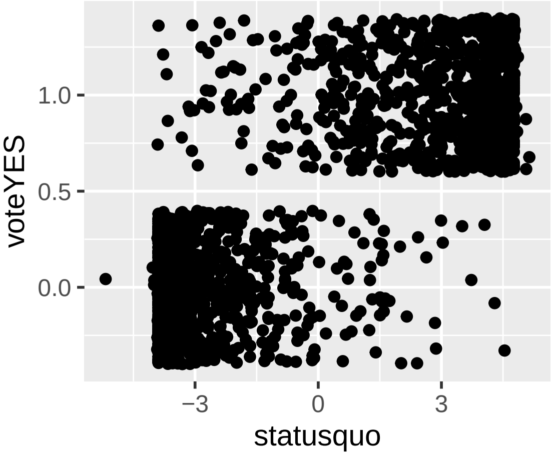
- If you want to do a scatterplot with a binary response variable, like the one in class, you need to add some `jitter`!
- The basic syntax is:

```
ggplot(data = dataset,
       aes(x = indepvar, y = bindepvar)) +
  geom_jitter(height = amount_x_jitter,
              width = amount_y_jitter)
```

- You need to change the `dataset`, the `indepvar`, and the `bindepvar`, the `amount_x_jitter`, and `amount_y_jitter` for a number between zero and one.
- It is great for a binary response variable!

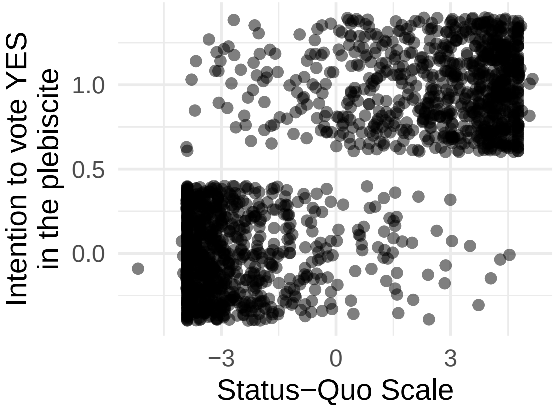
Scatter plots with binary dependent

```
ggplot(data = chilesurv, aes(x = statusquo, y = voteYES)) +
  geom_jitter(height = 0.4, width = 0)
```



Scatter plots with binary dependent

```
ggplot(data = chilesurv, aes(x = statusquo, y = voteYES)) +
  geom_jitter(fill = 'lightblue', alpha = 0.5, height = 0.4, width = 0) +
  labs(title = '', y = 'Intention to vote YES\nin the plebiscite',
       x = 'Status-Quo Scale') + theme_minimal()
```



Correlation

Correlation

- Computing correlations in R is very easy. The basic syntax is:

```
cor(dataset$var1, dataset$var2)
```

- You need to change the `dataset`, the `var1`, and the `var2`.
- By the way, the order of variables in the correlation does not matter.

Correlation

- Correlation between education expenditure and income:

```
cor(educexp$education, educexp$income)
## [1] 0.6675773
```

- **Your turn:** what is the correlation between education expenditure and the proportion of young people?

Bivariate regression

Bivariate regression

- Bivariate regression is to fit the function $Y = \beta_0 + \beta_1 X + \varepsilon$. The syntax is simple:

```
lm(Y ~ X, data = dataset)
```

- You need to change the `dataset`, the `Y` variable, and the `X` variable.
- `Y ~ X` is called the formula for your regression.
- It spits out a pair of numbers for β_0 (we call $\widehat{\beta}_0$) and β_1 (we call $\widehat{\beta}_1$).

Bivariate regression

- A bivariate regression for education expenditure explained by income:

```
lm(education ~ income, data = educexp)
##
## Call:
## lm(formula = education ~ income, data = educexp)
##
## Coefficients:
## (Intercept)       income
##    17.71003      0.05538
```

- **Your turn:** what are the bivariate regression estimates for education expenditure and the proportion of young people?

Today's Lab

- Barplots
- Violinplots
- Scatterplots
- Correlation
- Bivariate Regression

Next Lab

- More plots and analysis
- Some data wrangling

Questions?

See you in the next lab!