Crash Course on Data Analytics

Assoc Prof Peter Julian Cayton, PhD

2024-09-23

Flow of Presentation

1. Quick Introduction to R/RStudio
2. A Typical Data Analytics Workflow
3. A Crash Course Demonstration
4. Case Study: Collaborative Data Science and COVID-19
5. Further Sources and References

Materials Available Online

They may be found in the website below:

https://github.com/pacayton/Crash_course_On_Data_Analytics

Quick Introduction to R/RStudio

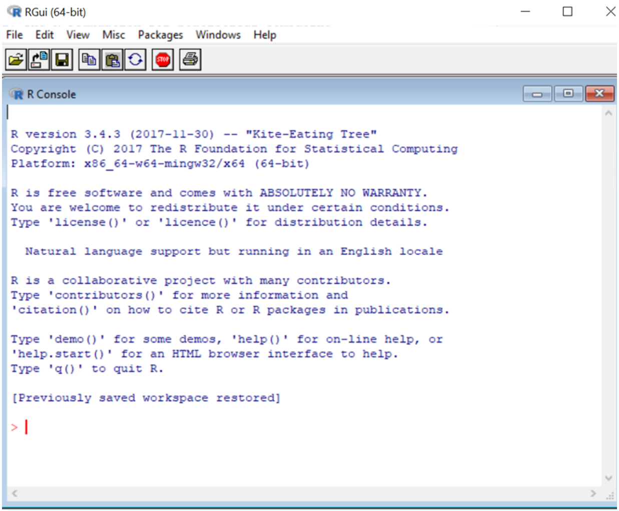
Quick Introduction to R/RStudio

R Programming Language

- R is a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. (source: The Comprehensive R Archive Network)
- Website: https://cran.r-project.org/

Quick Introduction to R/RStudio

R Programming Language

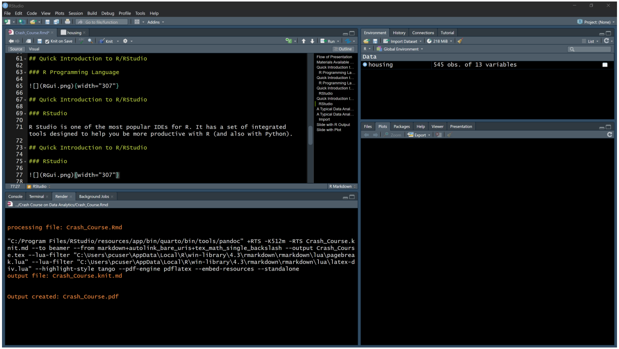


```
R version 3.4.3 (2017-11-30) -- "Kite-Eating Tree"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> 
```

Quick Introduction to R/RStudio

RStudio

- RStudio is one of the most popular IDEs for R. It has a set of integrated tools designed to help you be more productive with R (and also with Python).
- Website: https://posit.co/downloads/



```
## Quick Introduction to R/RStudio

### R Programming Language

![](RGui.png){width="307"}

## Quick Introduction to R/RStudio

### RStudio

R Studio is one of the most popular IDEs for R. It has a set of integrated
tools designed to help you be more productive with R (and also with Python).

## Quick Introduction to R/RStudio

### RStudio

![](RGui.png){width="307"}
```

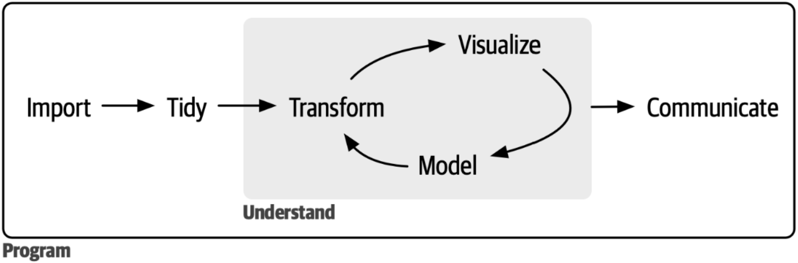
.../Crash Course on Data Analytics/Crash_Course.Rmd

```
processing file: Crash_Course.Rmd

"C:/Program Files/RStudio/resources/app/bin/quarto/bin/tools/pandoc" +RTS -K512m -RTS Crash_Course.k
nit.md --to beamer --from markdown+autolink_bare_uris+tex_math_single_backslash --output Crash_Cours
e.tex --lua-filter "C:\Users\pcuser\AppData\Local\R\win-library\4.3\rmarkdown\rmarkdown\lua\pagebrea
k.lua" --lua-filter "C:\Users\pcuser\AppData\Local\R\win-library\4.3\rmarkdown\rmarkdown\lua\latex-d
iv.lua" --highlight-style tango --pdf-engine pdflatex --embed-resources --standalone
output file: Crash_Course.knit.md


Output created: Crash_Course.pdf
```

```
housing             545 obs. of 13 variables
```

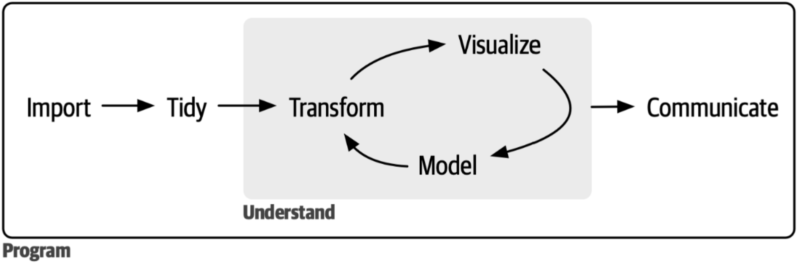
A Typical Data Analytics Workflow

A Typical Data Analytics Workflow



Source: Wickham, H and Grolemund, G (2017). R for Data Science. O'Reilly. https://r4ds.had.co.nz/introduction.html

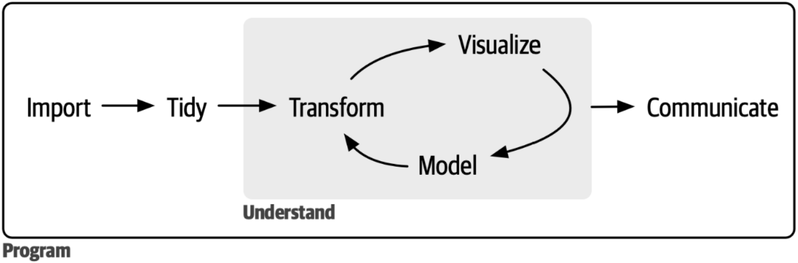
A Typical Data Analytics Workflow



Import

- Extracting the data from an internal database, a file, an online website, or thru a web application programming interface (API), to be loaded in R/RStudio

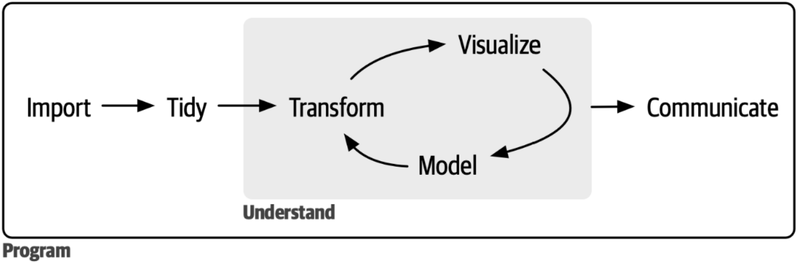
A Typical Data Analytics Workflow



Tidy

- Arranging the data into a neat data structure, with variable as columns and data points as rows.
- Included in this step would be data cleaning, data augmentation, missing data imputation, and others

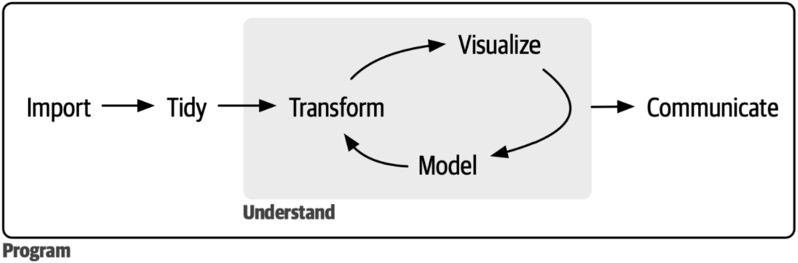
A Typical Data Analytics Workflow



Understand

- Generally, the steps to extract insights from data after tidying up.

A Typical Data Analytics Workflow

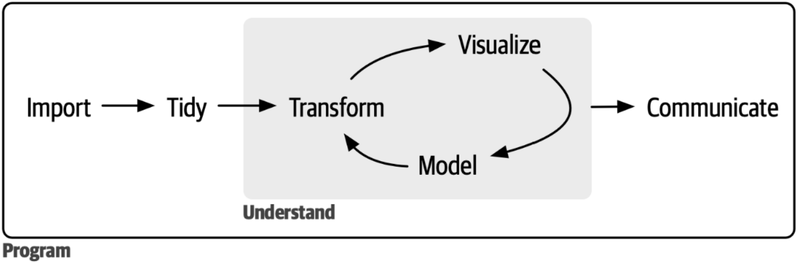


Transform

- Processing the data in preparation for further steps. Examples are:

1. Narrowing the data, e.g., by region or by age,
2. Computing new variables, e.g., length of days until recovery, or delays in reporting cases
3. Aggregating data, e.g., counting cases or solving rates/means

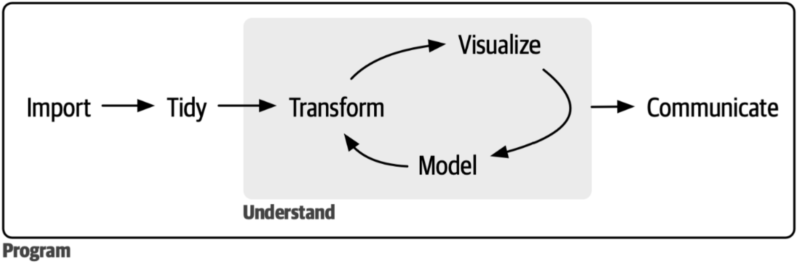
A Typical Data Analytics Workflow



Visualize

- Plot data into graphs so that patterns and features may be explored and insights be extracted from what is seen.

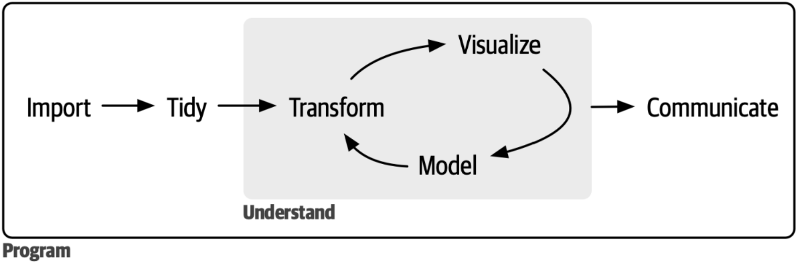
A Typical Data Analytics Workflow



Model

- When necessary, models help in summarizing the complex relationships and the patterns found from visualizations
- Designing models for prediction or forecasting

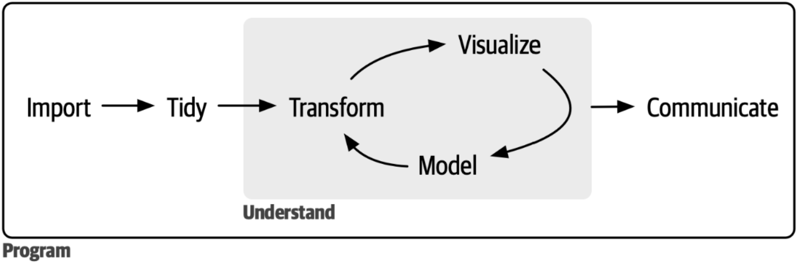
A Typical Data Analytics Workflow



Communicate

- Writing reports, creating dashboards, making presentations, compilations, etc.

A Typical Data Analytics Workflow



Program

- All these processes to be encapsulated in a data science project plan
- Possible to be encapsulated in one software, but it’s not impossible to use more than one depending on team members’ capabilities to process and analyze data.

A Crash Course Demonstration

A Crash Course Demonstration

Data for Demonstration:

M Yasser H. Housing Prices Dataset.
https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

A Crash Course Demonstration

Program

- upload the necessary packages
- pre-installation of packages should be done first

```
## Preamble Code

# install.packages("tidyverse", "knitr", "kableExtra",
#                  "moments", "stargazer")

## Packages to Use

library(tidyverse)
library(knitr)
library(kableExtra)
library(moments)
library(stargazer)
```

A Crash Course Demonstration

Program

- below is a set of codes for me to compute descriptive statistics later.

```
### generate basic stats

stat_vec <- function(x) {
  v <-as.numeric(x)
  vec <- c(mean(v, na.rm = TRUE),
           sd(v, na.rm = TRUE),
           skewness(v, na.rm = TRUE),
           kurtosis(v, na.rm = TRUE))
  return(vec)
}
```

A Crash Course Demonstration

Import

- get the dataset from an online repository (Github)
- location of the file: https://github.com/pacayton/Crash_course_On_Data_Analytics/raw/main/Housing.csv

```
housing <- read_csv("https://github.com/pacayton/Crash_cour
```

A Crash Course Demonstration

Import

```
housing[,1:3]
```

```
## # A tibble: 545 x 3
##       price  area bedrooms
##       <dbl> <dbl>    <dbl>
##  1 13300000  7420        4
##  2 12250000  8960        4
##  3 12250000  9960        3
##  4 12215000  7500        4
##  5 11410000  7420        4
##  6 10850000  7500        3
##  7 10150000  8580        4
##  8 10150000 16200        5
##  9  9870000  8100        4
## 10  9800000  5750        3
## # i 535 more rows
```

A Crash Course Demonstration

Import

```r
summary(housing[,1:3])
```

```
##      price               area          bedrooms
##  Min.   : 1750000   Min.   : 1650   Min.   :1.000
##  1st Qu.: 3430000   1st Qu.: 3600   1st Qu.:2.000
##  Median : 4340000   Median : 4600   Median :3.000
##  Mean   : 4766729   Mean   : 5151   Mean   :2.965
##  3rd Qu.: 5740000   3rd Qu.: 6360   3rd Qu.:3.000
##  Max.   :13300000   Max.   :16200   Max.   :6.000
```

```r
colnames(housing)
```

[1] “price” “area” “bedrooms” “bathrooms”
[5] “stories” “mainroad” “guestroom” “basement”
[9] “hotwaterheating” “airconditioning” “parking” “prefarea”
[13] “furnishingstatus”

A Crash Course Demonstration

Tidy

- fix the data for easier processing later
- ex: price_in_thousands & with_parking

```r
housing <- housing %>%
  mutate(
    price_in_thousands = price / 1000,
    with_parking = ifelse(parking > 0, TRUE, FALSE)
  )
```

A Crash Course Demonstration

Tidy

```
head(housing[,c("price", "parking",
                "price_in_thousands", "with_parking")])
```

```
## # A tibble: 6 x 4
##      price parking price_in_thousands with_parking
##      <dbl>   <dbl>              <dbl> <lgl>
## 1 13300000       2              13300 TRUE
## 2 12250000       3              12250 TRUE
## 3 12250000       2              12250 TRUE
## 4 12215000       3              12215 TRUE
## 5 11410000       2              11410 TRUE
## 6 10850000       2              10850 TRUE
```

A Crash Course Demonstration

Understand / Transform

- Computing basic statistics

```r
label <- c("Mean", "Standard Deviation",
           "Skewness", "Kurtosis")

comp_stats <- round(stat_vec(housing$price_in_thousands),
                    2)

stats_table <- rbind(label, comp_stats)
```

A Crash Course Demonstration

Understand / Transform

- Presenting basic statistics on a table

```
kable(t(stats_table), format="pipe")
```

label	comp_stats
Mean	4766.73
Standard Deviation	1870.44
Skewness	1.21
Kurtosis	4.93

A Crash Course Demonstration

Understand / Transform

- Presenting average house price by whether they are next to a main road (yes/no)

```
housing %>% group_by(mainroad) %>%
  summarise("Average House Price (in thousands)"
            = mean(price_in_thousands)) %>%
  kable(format="pipe")
```

mainroad	Average House Price (in thousands)
no	3398.905
yes	4991.777

A Crash Course Demonstration

Understand / Transform

- Presenting average house price by furnishing status (un/semi/furnished)

```r
housing %>% group_by(furnishingstatus) %>%
  summarise("Average House Price (in thousands)"
            = mean(price_in_thousands))%>%
  kable(format="pipe")
```

furnishingstatus	Average House Price (in thousands)
furnished	5495.696
semi-furnished	4907.524
unfurnished	4013.831

A Crash Course Demonstration

Understand / Transform

- Presenting average house price by parking availability (TRUE/FALSE)

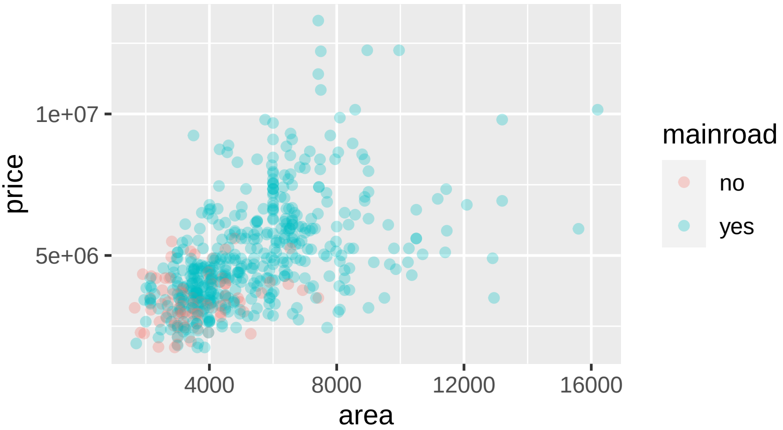
```r
housing %>% group_by(with_parking) %>%
  summarise("Average House Price (in thousands)"
            = mean(price_in_thousands))%>%
  kable(format="pipe")
```

with_parking	Average House Price (in thousands)
FALSE	4136.017
TRUE	5533.327

A Crash Course Demonstration

Understand / Visualize

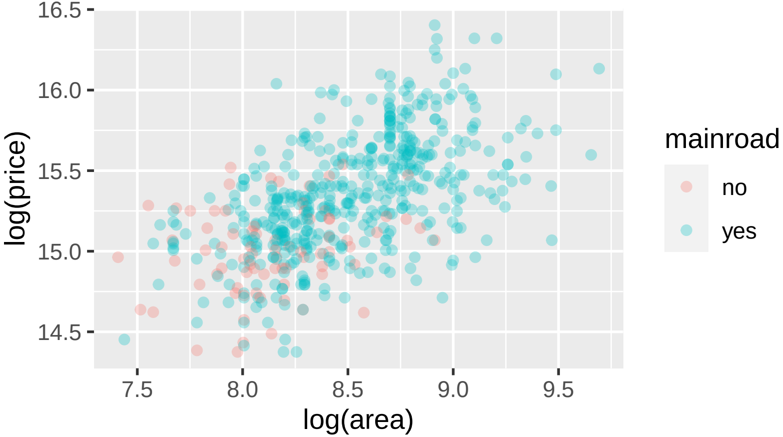
```r
ggplot(housing) +
  geom_point(aes(x = area, y = price,
                 color = mainroad), alpha = 0.3)
```



A Crash Course Demonstration

Understand / Visualize

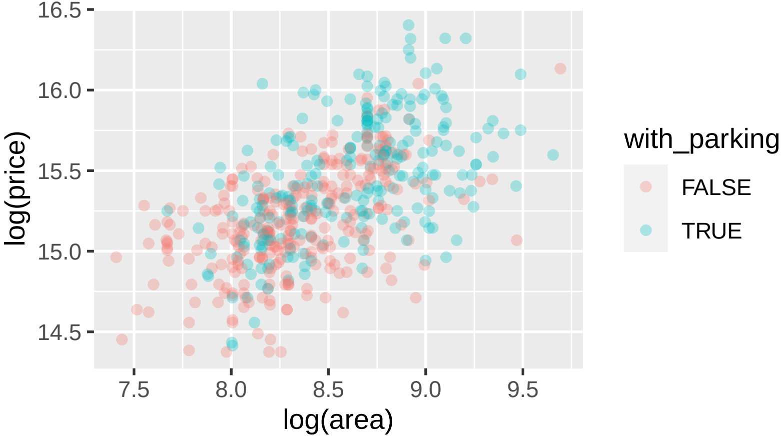
```
ggplot(housing) +
  geom_point(aes(x = log(area), y = log(price),
                 color = mainroad), alpha = 0.3)
```



A Crash Course Demonstration

Understand / Visualize

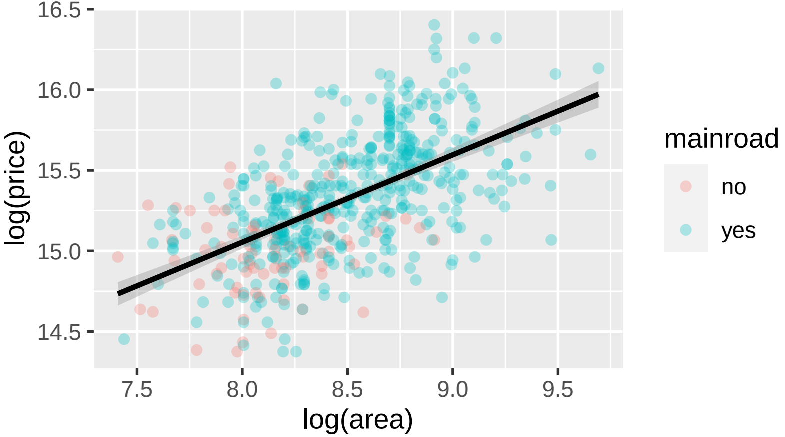
```
ggplot(housing) +
  geom_point(aes(x = log(area), y = log(price),
                 color = with_parking), alpha = 0.3)
```



A Crash Course Demonstration

Understand / Visualize

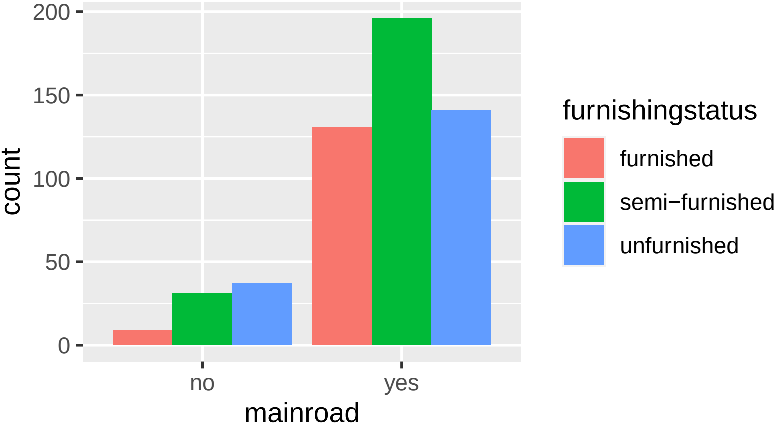
```
ggplot(housing, aes(x = log(area), y = log(price))) +
  geom_point(aes(color = mainroad), alpha = 0.3) +
  geom_smooth(method = "lm", color = "black")
```



A Crash Course Demonstration

Understand / Visualize

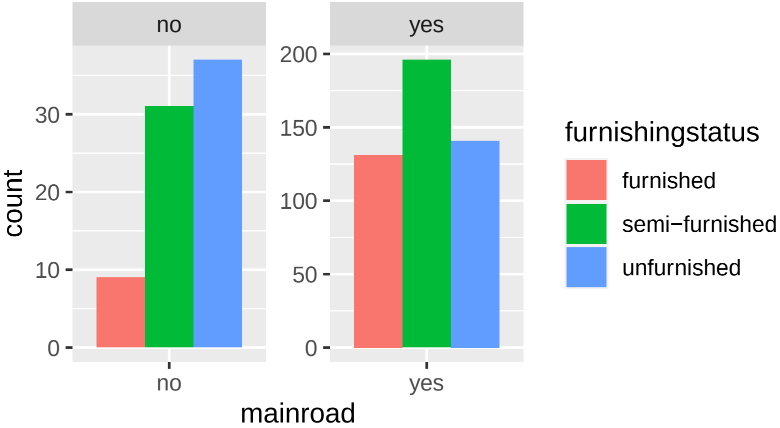
```r
ggplot(housing, aes(x = mainroad)) +
  geom_bar(aes(fill = furnishingstatus), stat = "count",
           position = "dodge")
```



A Crash Course Demonstration

Understand / Visualize

```r
ggplot(housing, aes(x = mainroad)) +
  geom_bar(aes(fill = furnishingstatus), stat = "count",
           position = "dodge") +
  facet_wrap(.~ mainroad, scale = "free")
```



A Crash Course Demonstration

Understand / Model

- we assume the following:

$$Price_{\text{in thousands}} = \beta_0 + \beta_1 \times area + \beta_2 \times bedrooms + \epsilon$$

```r
linear_model <- lm(price_in_thousands ~ area + bedrooms,
                   data = housing)
```

A Crash Course Demonstration

Understand / Model

- the result is:

```r
stargazer(summary(linear_model)$coefficients,
          type = "text")
```

```
##
## ===================================================
##             Estimate Std. Error t value Pr(> | t| )
## ---------------------------------------------------
## (Intercept) 391.126   287.351    1.361     0.174
## area         0.424     0.030    14.260       0
## bedrooms    739.566    87.381    8.464       0
## ---------------------------------------------------
```

$$PredictedPrice_{\text{in thousands}} = 391.126 + 0.424 \times area + 739.566 \times bedrooms$$

A Crash Course Demonstration

Communicate

- The current slide presentation is produced within the RStudio interface, helping with both writing reports and running statistical procedures in 1 code file!

Case Study: Collaborative Data Science and COVID-19

Case Study: Collaborative Data Science and COVID-19

You can access the case study handout here:

https://github.com/pacayton/Crash_course_On_Data_Analytics/blob/main/Collaborative%20Data%20Science%20and%20COVID-19.pdf

Further Sources and References

Further Sources and References

Offered Trainings

- Philippine Statistical Research and Training Institute: https://psrti.gov.ph/ (Website currently under maintenance) or https://www.facebook.com/PSRTI.Official
- UP Statistical Center Research Foundation: For more details on trainings offered, follow https://www.facebook.com/UPDStat

Online Reference on Data Science using R/RStudio

- R for Data Science (2e): https://r4ds.hadley.nz/

Thank you very much and have a great day!