The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

G53FIV: Fundamentals of Information Visualization

Lecture 7: Visualization with R – Advanced

Ke Zhou
School of Computer Science
Ke.Zhou@nottingham.ac.uk

https://moodle.nottingham.ac.uk/course/view.php?id=68644

Last Lecture

Visualization with R

R is a tool for...

Data Manipulation

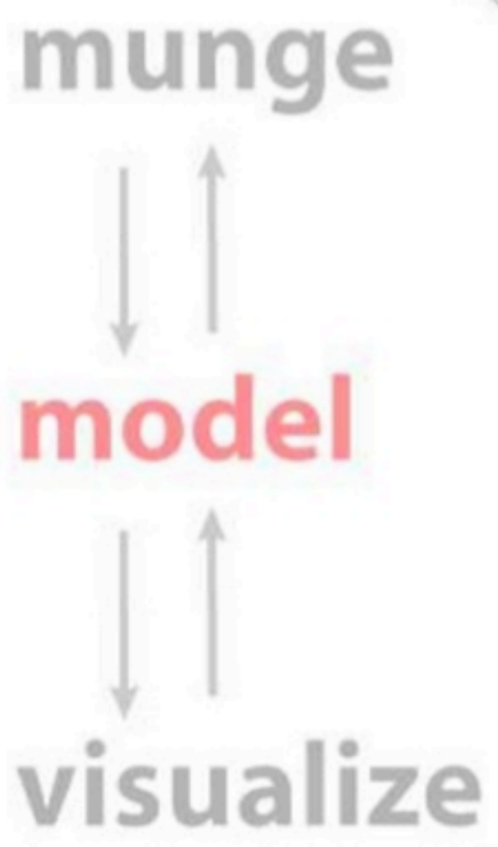
- connecting to data sources
- slicing & dicing data

Modeling & Computation

- statistical modeling
- numerical simulation

Data Visualization

- visualizing fit of models
- composing statistical graphics



Building a Plot in ggplot2

data to visualize (a data frame)

map variables to ***aes***thetic attributes

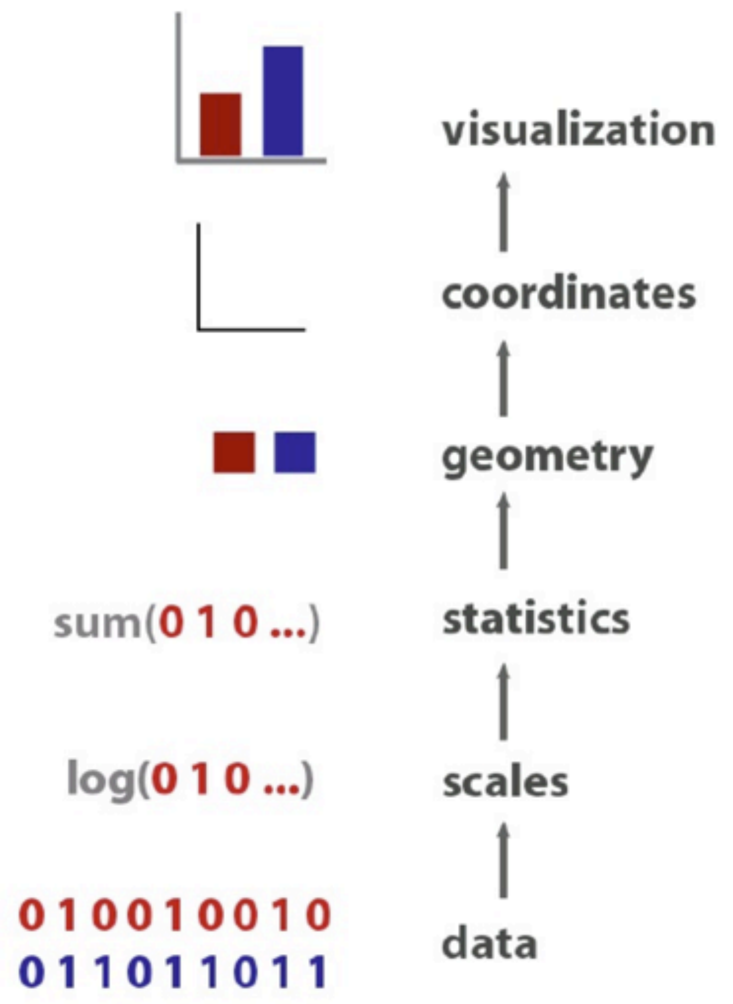
geometric objects – what you see (points, bars, etc)

scales map values from data to aesthetic space

faceting subsets the data to show multiple plots

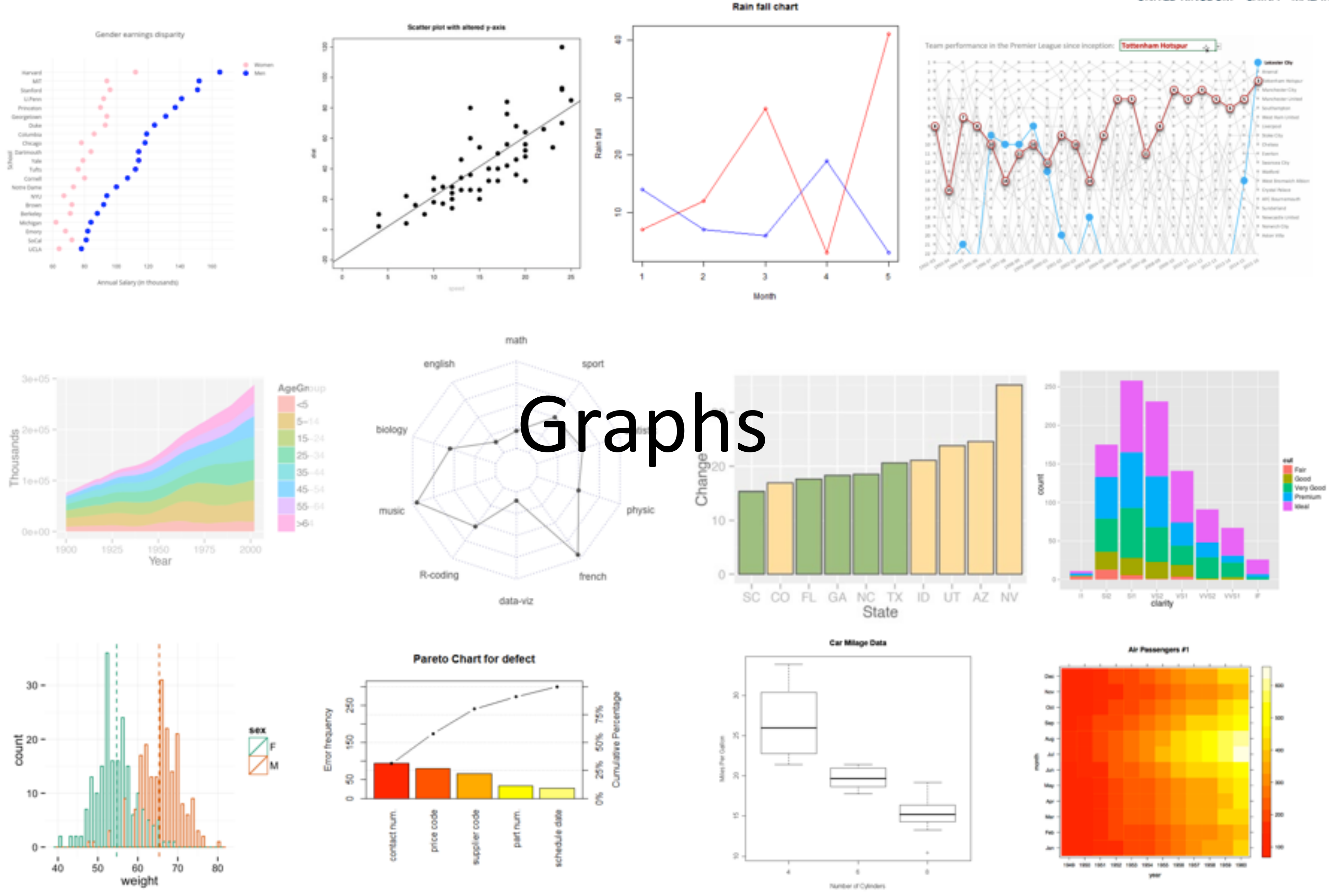
statistical transformations – summarize data

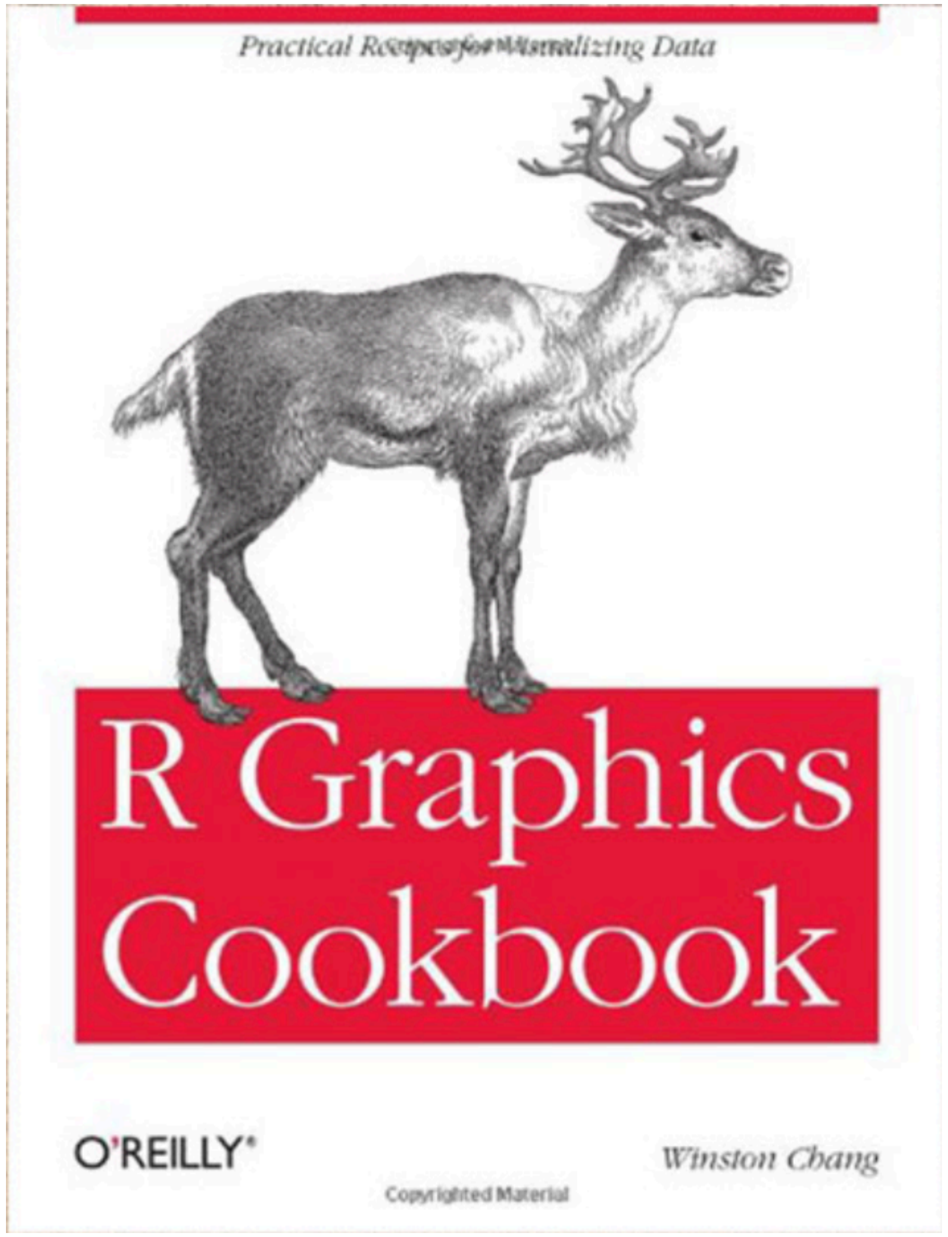
coordinate systems put data on plane of graphic



Wickham 2009

Graphs


Gender earnings disparity
Scatter plot with altered y-axis
Rain fall chart
Month
Team performance in the Premier League since inception:
Tottenham Hotspur
AgeGroup
Thousands
Year
math
sport
physic
french
data-viz
R-coding
music
biology
english
Change
State
SC
CO
FL
GA
NC
TX
ID
UT
AZ
NV
count
clarity
Fair
Good
Very Good
Premium
Ideal
sex
F
M
weight
Pareto Chart for defect
Error frequency
Cumulative Percentage
contact num.
price code
supplier code
part num.
schedule date
Car Milage Data
Number of Cylinders
Air Passengers #1
year
month



Practical Recipes for Visualizing Data

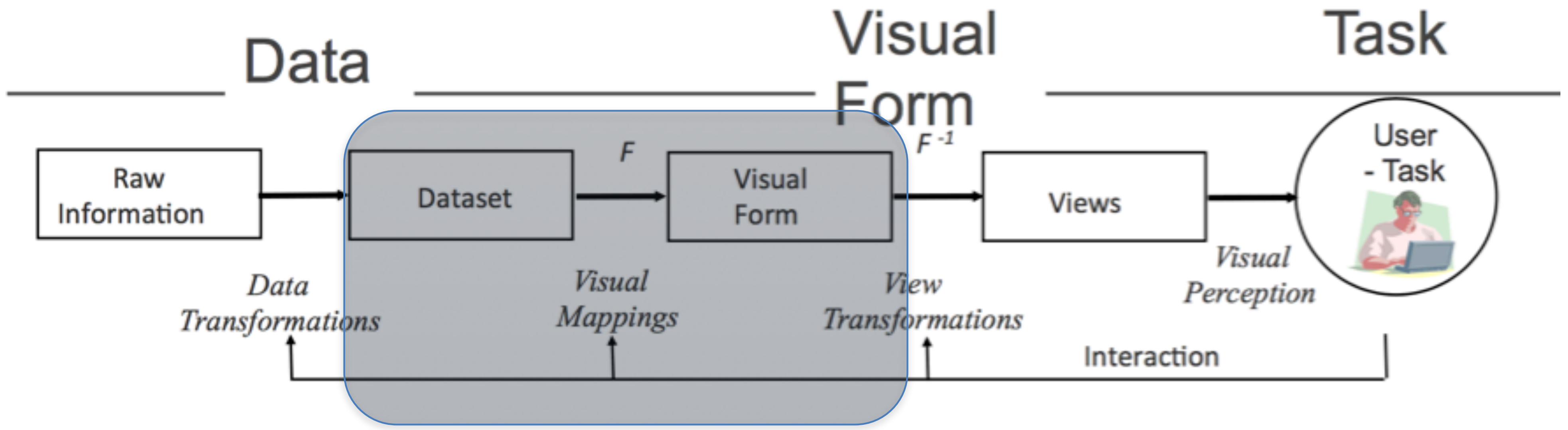
R Graphics Cookbook

O'REILLY®

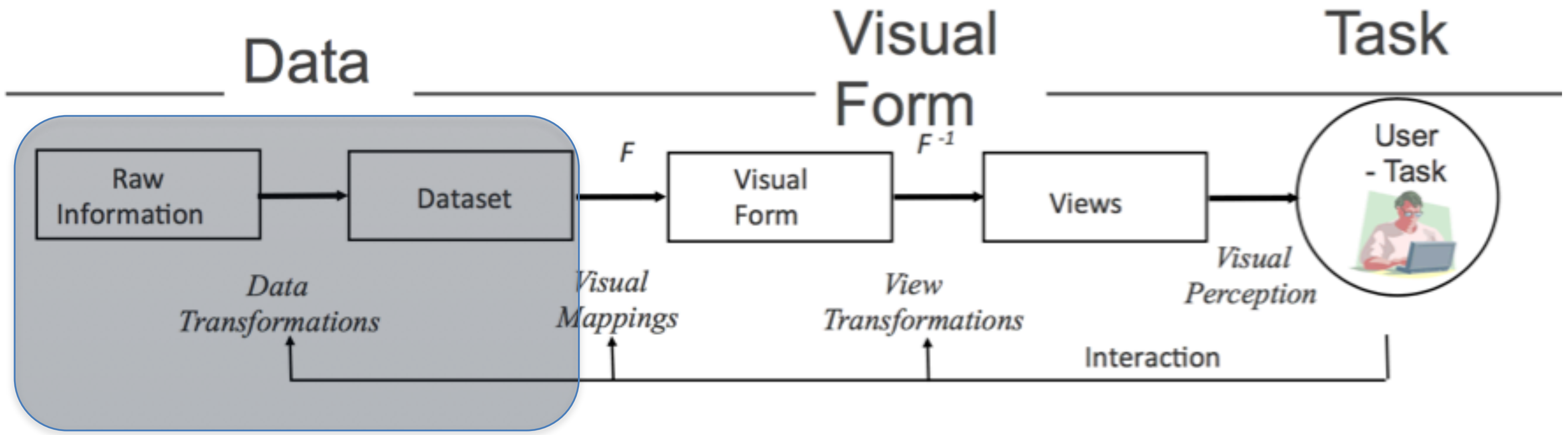
Winston Chang

Copyrighted Material

Seven Stages of Visualization



Seven Stages of Visualization



G53FIV Early Module Feedback

- Survey (on Moodle)
  - Anonymous
  - Seeking for constructive feedback
  - Two compulsory multiple questions
  - A few open-ended questions
    - If you have specific feedback or suggestions for improvement.

- A summary of the survey (and action points for improvements) will be presented.

Overview

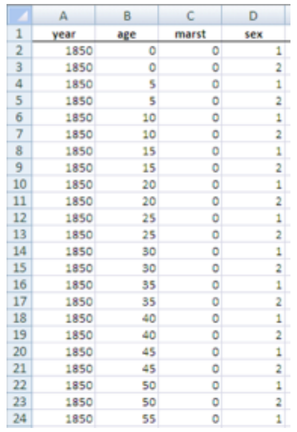
- Data Manipulations with R
- Brief (Coursework) Case Study with R

Data Manipulations with R

Dr. Ke Zhou (http://www.cs.nott.ac.uk/~pszkz/)

Transform Data: A Swiss-Army Knife

- Indexing
- Three ways to index into a data frame
  - Array of integer indices
  - Array of character names
  - Array of logical Booleans
- Examples:
  - **df[1:3,]**
  - **df[c(“New York”, “Chicago”),]**
  - **df[c(TRUE, FALSE, TRUE, TRUE),]**



	A	B	C	D
1	year	age	marst	sex
2	1850	0	0	1
3	1850	0	0	2
4	1850	5	0	1
5	1850	5	0	2
6	1850	10	0	1
7	1850	10	0	2
8	1850	15	0	1
9	1850	15	0	2
10	1850	20	0	1
11	1850	20	0	2
12	1850	25	0	1
13	1850	25	0	2
14	1850	30	0	1
15	1850	30	0	2
16	1850	35	0	1
17	1850	35	0	2
18	1850	40	0	1
19	1850	40	0	2
20	1850	45	0	1
21	1850	45	0	2
22	1850	50	0	1
23	1850	50	0	2
24	1850	55	0	1

Transform Data: A Swiss-Army Knife

- subset – extract subsets meeting some criteria

```
subset(Insurance, District==1)
subset(Insurance, Claims < 20)
```

- transform – add or alter a column of a data frame

```
transform(Insurance, Propensity=Claims/Holders)
```

- cut – cut a continuous value into groups

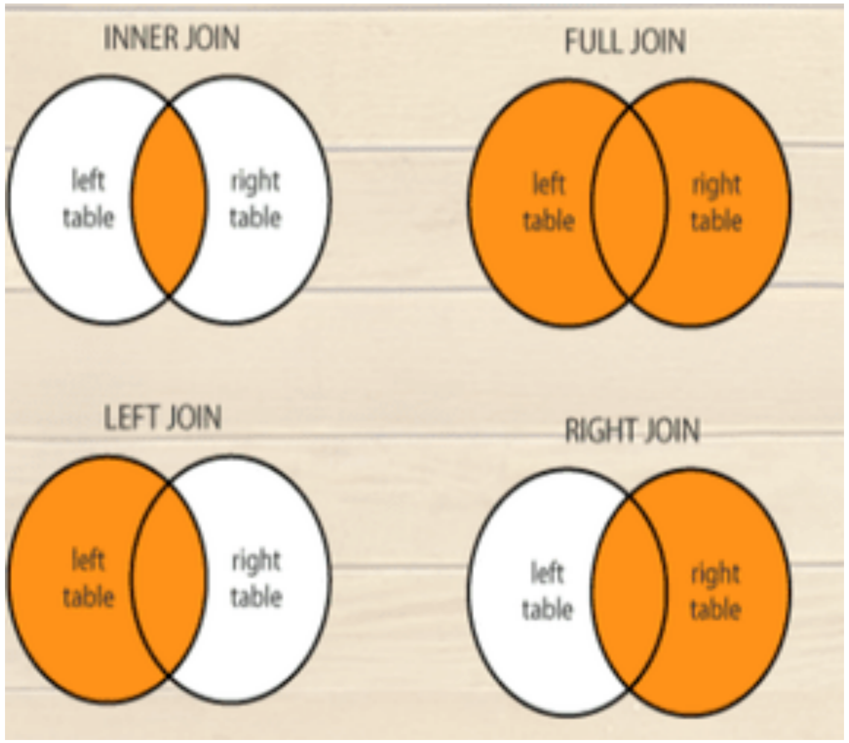
```
cut(Insurance$Claims, breaks=c(-
1,100,Inf), labels=c('lo','hi'))
```

- Put it all together: create a new, transformed data frame

```
transform(subset(Insurance, District==1),
   ClaimLevel=cut(Claims, breaks=c(-1,100,Inf),
   labels=c('lo','hi')))
```

Joining Two Data Frames

- inner_join(df1, df2, by = "common_column")
- ?join
  - Left_join, right_join
  - Inner_join, outer_join
- merge(x=df1, y=df2, by.x="id", by.y="bid")



More about Data Manipulations

- Packages
  - plyr
  - data.table
  - reshape2
  - doBY
  - sqldf
  - and many more

The University of Nottingham
UNITED KINGDOM · CHINA · MALAYSIA

dplyr: A Grammar of Data Manipulation

- Very intuitive, once you understand the basics
- Very fast
  - Created with execution times in mind
- Easy for those migrating from the SQL world
- When written well, your code reads like a "recipe"
- "Code the way you think"

https://cran.rstudio.com/web/packages/dplyr/vignettes/introduction.html

Pipe Operator

- Library(maggritr)
  - A R package launched on Jan 2014
  - A “magic” operator called the PIPE was introduced
  - %>%
  - i.e. “AND THEN”, “PIPE TO”

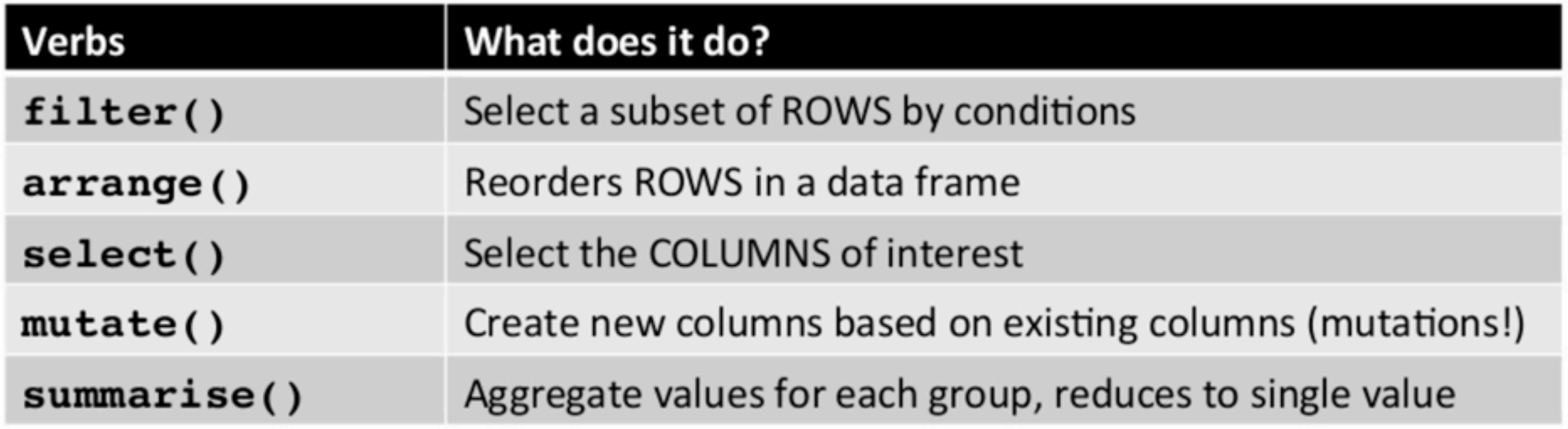
```
round(sqrt(1000), 3)

library(magrittr)
1000 %>% sqrt %>% round()
1000 %>% sqrt %>% round(.,3)
```

Take 1000, and then its sqrt
And then round it

1000
Sqrt function
31.62278
Round function

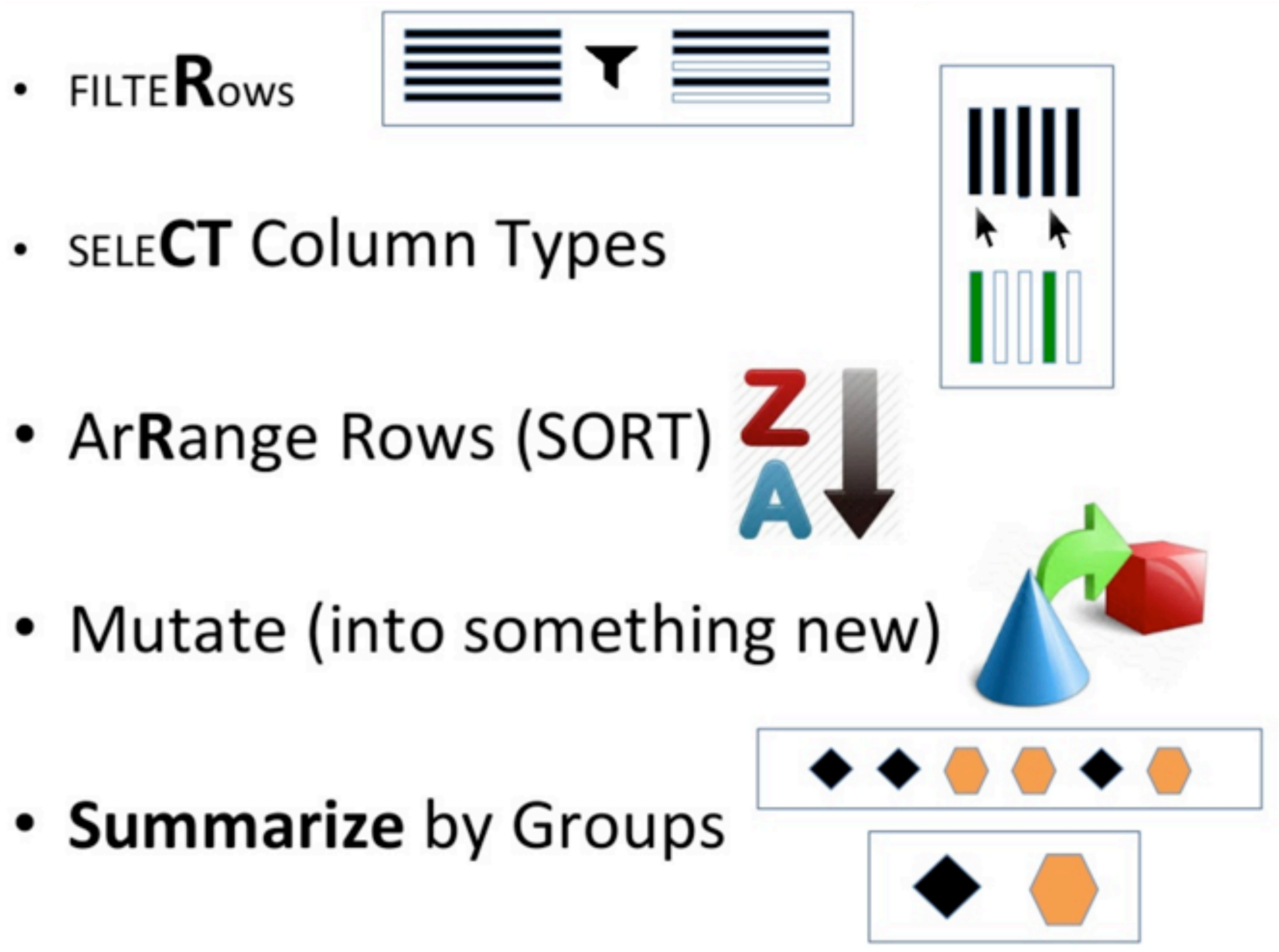
dplyr

- dplyr takes the %>% operator and uses it to great effect for manipulating data frames
  - Works only with data frames
  - 5 basic “verbs” work for 90% of data manipulations



Verbs	What does it do?
`filter()`	Select a subset of ROWS by conditions
`arrange()`	Reorders ROWS in a data frame
`select()`	Select the COLUMNS of interest
`mutate()`	Create new columns based on existing columns (mutations!)
`summarise()`	Aggregate values for each group, reduces to single value

5 Basic Verbs



- FILTE**R**ows
- SELE**CT** Column Types
- Ar**R**ange Rows (SORT)
- Mutate (into something new)
- **Summarize** by Groups

Movies dataset

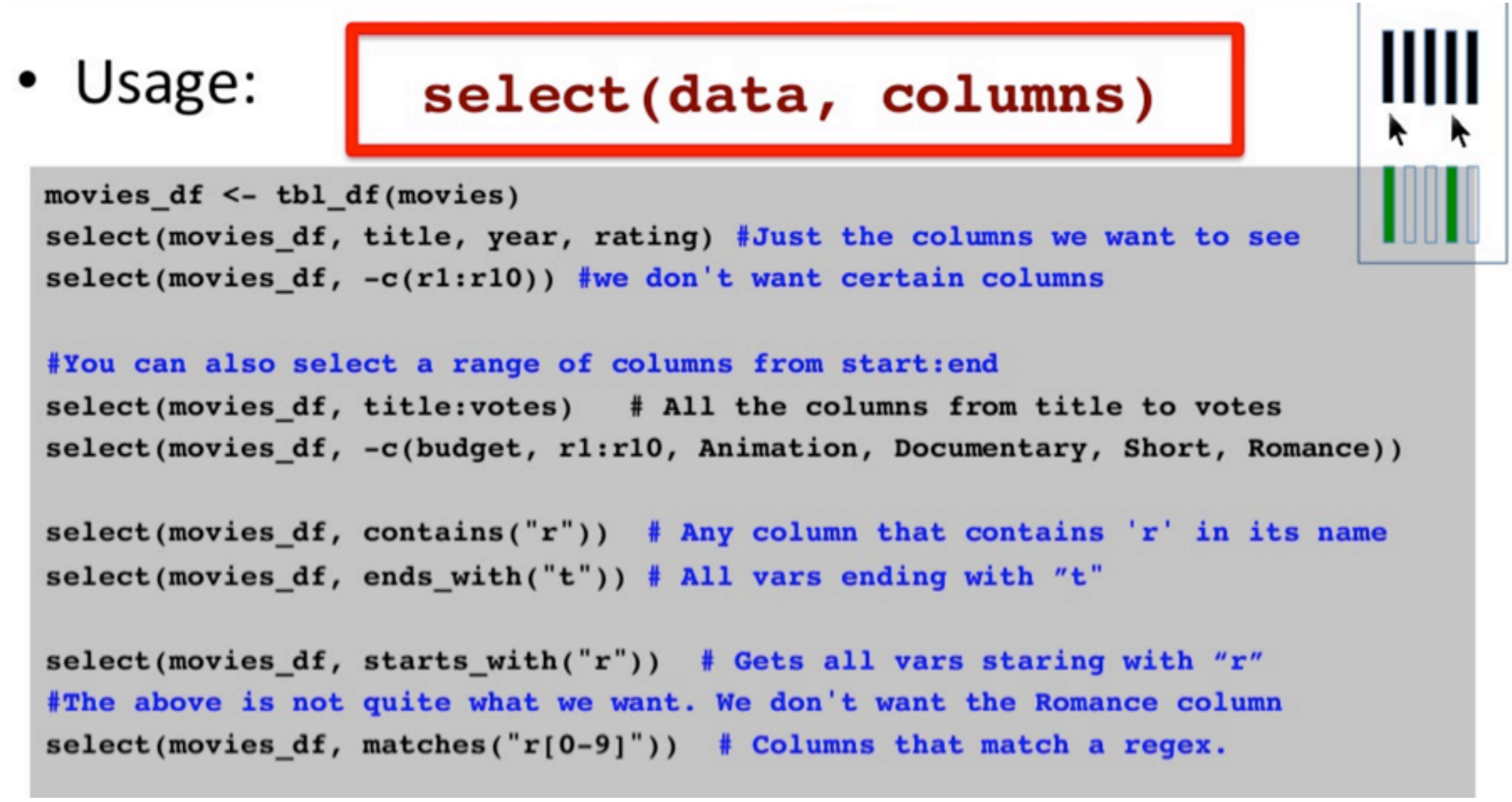
title	year	budget	votes	length	Documentary	rating	…
Titanic	1997	200,000,000	1000	195	0	7.8	…
Leon	1994	16,000,000	500	90	0	8.6	…
McQueen	2018	52,000,000	200	91	1	7.9	…

Filter()

- Usage: `filter(data, condition)`
  - Returns a subset of rows
  - Multiple conditions can be supplied.
  - They are combined with an AND

```
movies_with_budgets <- filter(movies_df, !is.na(budget))
filter(movies, Documentary==1)
filter(movies, Documentary==1) %>% nrow()
good_comedies <- filter(movies, rating > 9, Comedy==1)
dim(good_comedies) #171 movies

#' Let us  say we only want highly rated comdies, which a lot
of people have watched, made after year 2000.
movies %>%
  filter(rating >8, Comedy==1, votes > 100, year > 2000)
```

Select()



- Usage:

```
select(data, columns)
```

```
movies_df <- tbl_df(movies)
select(movies_df, title, year, rating) #Just the columns we want to see
select(movies_df, -c(r1:r10)) #we don't want certain columns

#You can also select a range of columns from start:end
select(movies_df, title:votes)   # All the columns from title to votes
select(movies_df, -c(budget, r1:r10, Animation, Documentary, Short, Romance))

select(movies_df, contains("r"))  # Any column that contains 'r' in its name
select(movies_df, ends_with("t")) # All vars ending with ”t"

select(movies_df, starts_with("r"))  # Gets all vars staring with “r”
#The above is not quite what we want. We don't want the Romance column
select(movies_df, matches("r[0-9]"))  # Columns that match a regex.
```

Arrange()

Usage:

```
arrange(data, column_to_sort_by)
```

– Returns a reordered set of rows

– Multiple inputs are arranged from left-to-right

```
movies_df <- tbl_df(movies)
arrange(movies_df, rating) #but this is not what we want
arrange(movies_df, desc(rating))
#Show the highest ratings first and the latest year…
#Sort by Decreasing Rating and Year
arrange(movies_df, desc(rating), desc(year))
```

What's the difference between these two?

```
arrange(movies_df, desc(rating), desc(year))
arrange(movies_df, desc(year), desc(rating))
```

Mutate()

- Usage:

```
mutate(data, new_col = func(oldcolumns)
```

- Creates new columns, that are functions of existing variables

```
mutate(iris, aspect_ratio = Petal.Width/Petal.Length)

movies_with_budgets <- filter(movies_df, !is.na(budget))
mutate(movies_with_budgets, costPerMinute = budget/length) %>%
  select(title, costPerMinute)
```

Group_by() and Summarize()

```
group_by(data, column_to_group) %>%
summarize(function_of_variable)
```

- Group_by creates groups of data
- Summarize aggregates the data for each group

```
by_rating <- group_by(movies_df, rating)

by_rating %>% summarize(n())

avg_rating_by_year <-
    group_by(movies_df, year) %>%
    summarize(avg_rating = mean(rating))
```

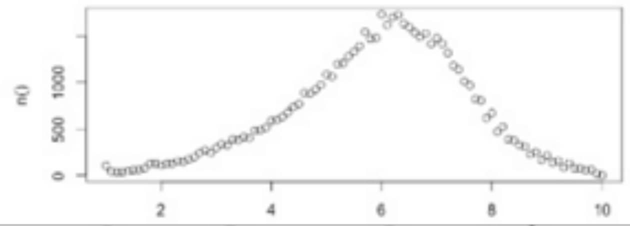
Chain the "Verbs" Together

- Chain them together

```
producers_nightmare <-
  filter(movies_df, !is.na(budget)) %>%
  mutate(costPerMinute = budget/length) %>%
  arrange(desc(costPerMinute)) %>%
  select(title, costPerMinute)
```

- Can also be fed to a "plot" command

```
movies %>%
  group_by(rating) %>%
  summarize(n()) %>%
  plot() # plots the histogram of movies by Each value of rating
```



Practice

- Find all the post-2000 comedy movies with over 1,000,000 budget, rank them by rating in the decreasing order, and output their title and rating

A Case Study:
House Price Visualization

All materials are available on Moodle.

Pick a dataset of your interest

- https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads
- Our Price Paid Data includes information on all property sales in England and Wales that are sold for full market value and are lodged with us for registration.
- Format:
  - tid,price,date,postcode,type,age,tenure,paon,saon,street,locality,city,district,county,ppdcategory,recordstatus

Pose the initial questions (3 to 5) that you would like to answer

- RQ1: When do property sales generally happen? Which month is the best month to sell?
- RQ2: How do property sales change according to different property types and age?
- RQ3: How do property sale price compare across different counties?

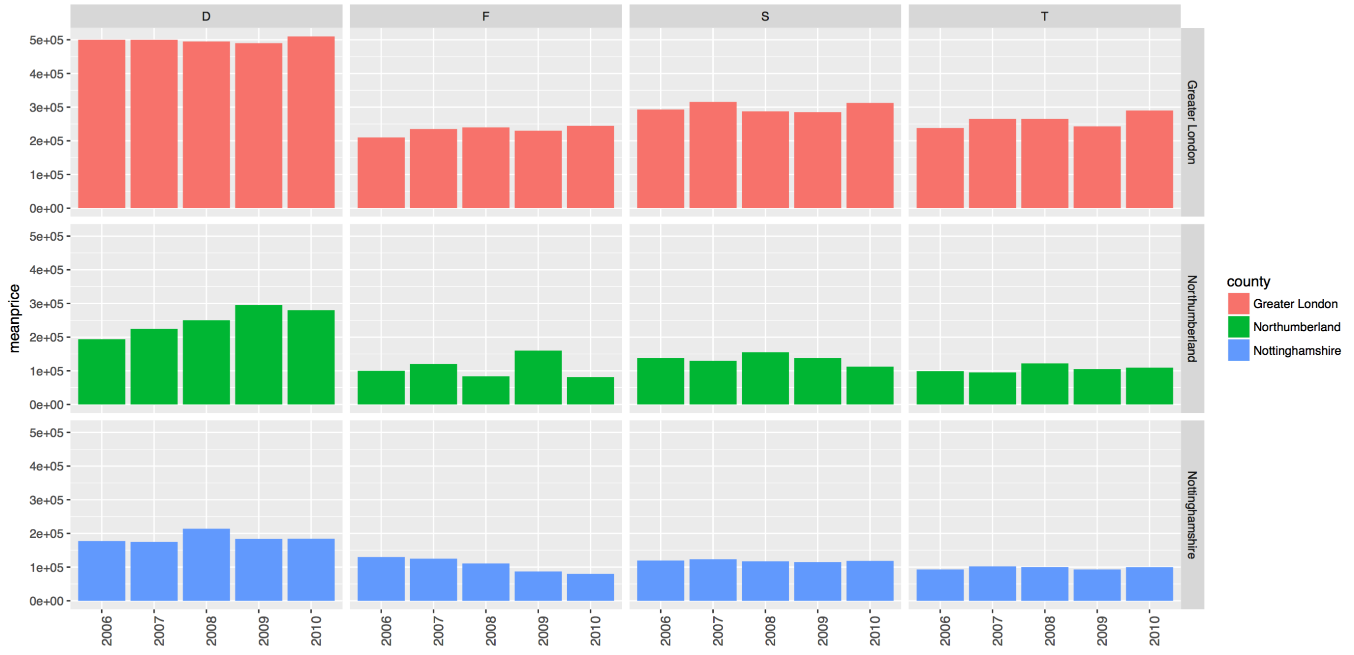
Assess the fitness of the data

- Can we answer all those questions by using only the land registry data?

- RQ3: How do property sale price compare across different counties?
  - Although we have area data, we need the county data, in order to compare across different counties.
  - County-postcode mapping data:
    - https://github.com/Gibbs/uk-postcodes/blob/master/postcodes.csv

Answer the initial questions by visualizing the dataset using R

- Demo
- Load data
- Manipulate data
- Visualization



Further refine/propose questions

- Example question
  - RQ4: Which types of properties and in which county suffer the most from the economical crisis?
- How to manipulate/process data?
  - E.g. need to calculate year (month) over year (month) change
- What visualization techniques to use? What if there are too many variables?

Course Work Deadline

- April 8th 2018
- Written report
  - 3000 words, max 10 pages
  - R codes as well

Next Lecture

- Topic:
  - Visualization Tools and Visual Perception