MATH 300 : : CHEAT SHEET

Data

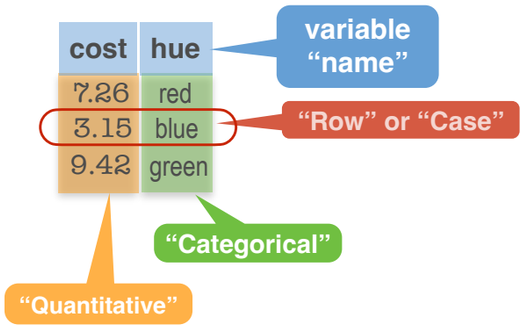
Essentials:
What is the "unit of observation?" All the rows represent a particular "kind of thing," e.g. a person or a flight or a book.

What are the variable names and types?
- Quantitative (numerical)
- Categorical (names of levels)

Data Frames

Every data frame has a name, such as `flights`, `Galton`, `Hill_racing`

Give a name to the data frames you create with the assignment operator
`My_new_frame <- Original %>% …`



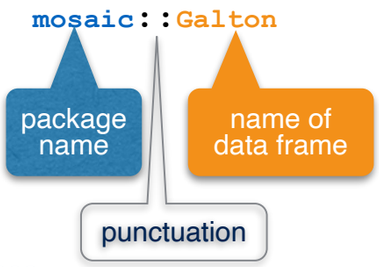
R Packages

An R "package" is a unit for the distribution of software. Packages we will use include
`mosaic`, `dplyr`, `ggplot2`, `math300`

In each Rmd file, you must "load" the packages that are used for that file, e.g. `library(dplyr)`

Most of the data in the course will come from packages. Once the package is loaded, you can refer to the data frame by its name, and get documentation with `?[name]`.

Sometimes we will refer to the package explicitly by using the "double colon" notation, e.g.



Databases

A "database" is a collection of related data frames, typically each with its own "unit of observation." Example: the `nycflights13` package contains a database consisting of `flights`, `airlines`, `weather`, `planes`, `airports`

Such "relational databases" are used throughout the economy to store complex, structured data.

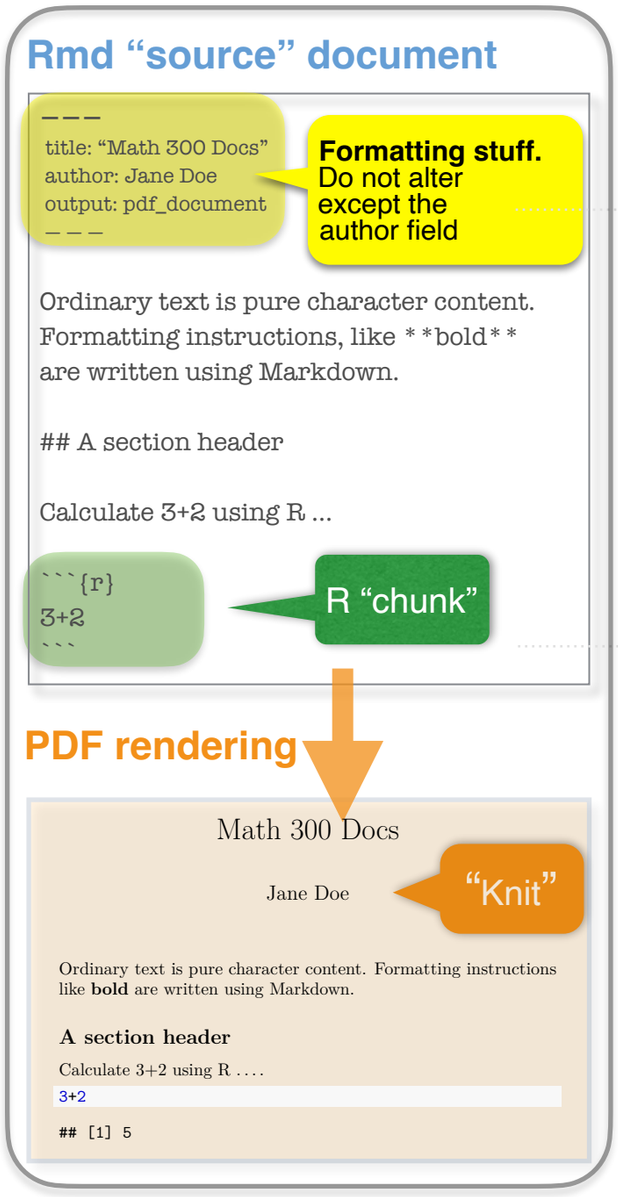
Class Documents

Students edit two kinds of documents, both in RMD format, both available through Posit.cloud in the Math 300 [semester] space.

Lesson Notes for every class day
- Typical name: *Lesson Notes/Lesson 3*
- Not graded. Instructors can access your RMD file directly as need be.

Problem Sets: two in each of the four blocks of the course
- Typical name: *Problem Sets/Problem set 1*
- Graded. Knit Rmd -> PDF and upload to GradeScope.

Rmd "source" document



Regression Modeling

The fundamental stat technique used in Math 300.

- *Response variable:* always quantitative.
- *Explanatory variable(s)*: Can be quantitative or categorical.
- *Specification*: A tilde expression, e.g.
  - `response ~ 1`
  - `response ~ ex1`
  - `response ~ ex1 + ex2`

Categorical response variable? Use zero_one(), e.g.

```
Galton <- Galton %>%
   mutate(
       sex_01 = zero_one(sex, one="F"))
```

Training on data ("fitting")

Takes as arguments:
(1) a data frame and (2) a model specification.

Returns:
A "model object" that can be graphed or summarized.

Usage:

```
Mod1 <- lm(height ~ mother,
           data=Galton)
```

model specification

data frame

your name for the model object

Model summaries

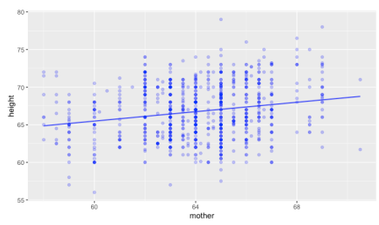
```
Mod1 %>% coef_summary()
Mod1 %>% conf_interval()
Mod1 %>% regression_summary()
Mod1 %>% anova_summary()
```

Graphing a model

Must provide variable for x axis!

```
Mod1 %>% model_plot(x = mother)
```

Optional arguments:
`interval='prediction'`
`interval='confidence'`



Logistic regression

Use when model output is a **probability**.

```
ModB <- glm(
  zero_one(outcome, one="Alive") ~
       age + smoker,
  data=Whickham,
  family="binomial")
```

ggplot Graphics

Every graphic involves two steps:

1. Define the x- and y-axes: the graphics frame

```
my_frame <- Galton %>%
    ggplot(aes(x=mother, y=father))
```

data frame
annoying bit
Variable names
graphics frame making function

2. Add graphics layers ("geoms") with +

```
my_frame + geom_point( )
my_frame + geom_jitter( )
my_frame + geom_violin( )
```

Optional arguments for geoms:
- transparency: `alpha = 0.5`
- color: `color="red"` or `aes(color=sex)`
- amount of jittering: `height=0.2, width=0.2`
- For geom_violin:
  `fill="blue", alpha=0.2, color=NA`

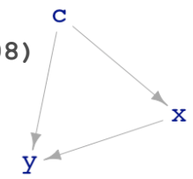
What's with aes() ?
Sets a graphic **aesthetic** to correspond with a *variable* as opposed to a *constant* like "red"

DAGs

Used to represent causal connections & to generate simulated data.

Example: `dag08`

```
dag_draw(dag08)
```




```
print(dag08)
c ~ exo()
x ~ c + exo()
y ~ x + c + 3 + exo()

sample(dag08, size=4)
# A tibble: 4 × 3
        c      x      y
    <dbl>  <dbl> <dbl>
1   0.764  0.355  4.36
2  -0.855 -0.711  2.12
3  -0.468 -1.69   1.40
4  -0.681  0.145  3.49
```

`exo( )` generates exogenous noise