

How to Teach a Workshop



Beginner



"I don't know what
I don't know."

Competent
Practitioner



"I can do it, but I may
look things up."

Expert



"I can handle anything
you throw at me"

Beginner



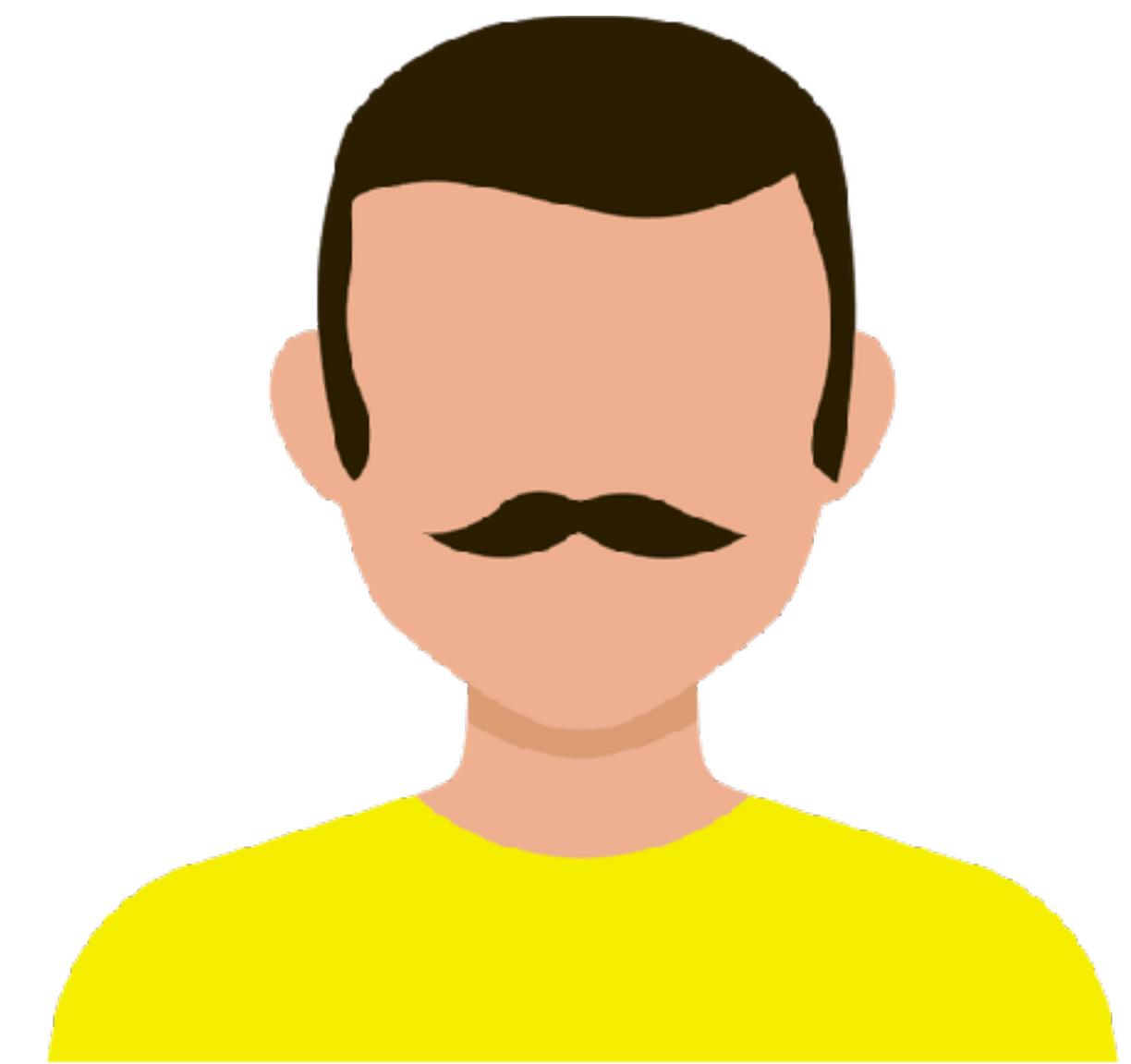
"I don't know what
I don't know."

Competent
Practitioner



"I can do it, but I may
look things up."

Expert



"I can handle anything
you throw at me"

Beginner



"I don't know what
I don't know."

Competent
Practitioner

Expert

"I can do it, but I may
look things up."

"I can handle anything
you throw at me"

Beginner

Competent
Practitioner

Expert



"I don't know what
I don't know."

"I can do it, but I may
look things up."

"I can handle anything
you throw at me"

Beginner



"I don't know what
I don't know."

Competent
Practitioner



"I can do it, but I may
look things up."

Expert



"I can handle anything
you throw at me"

Beginner



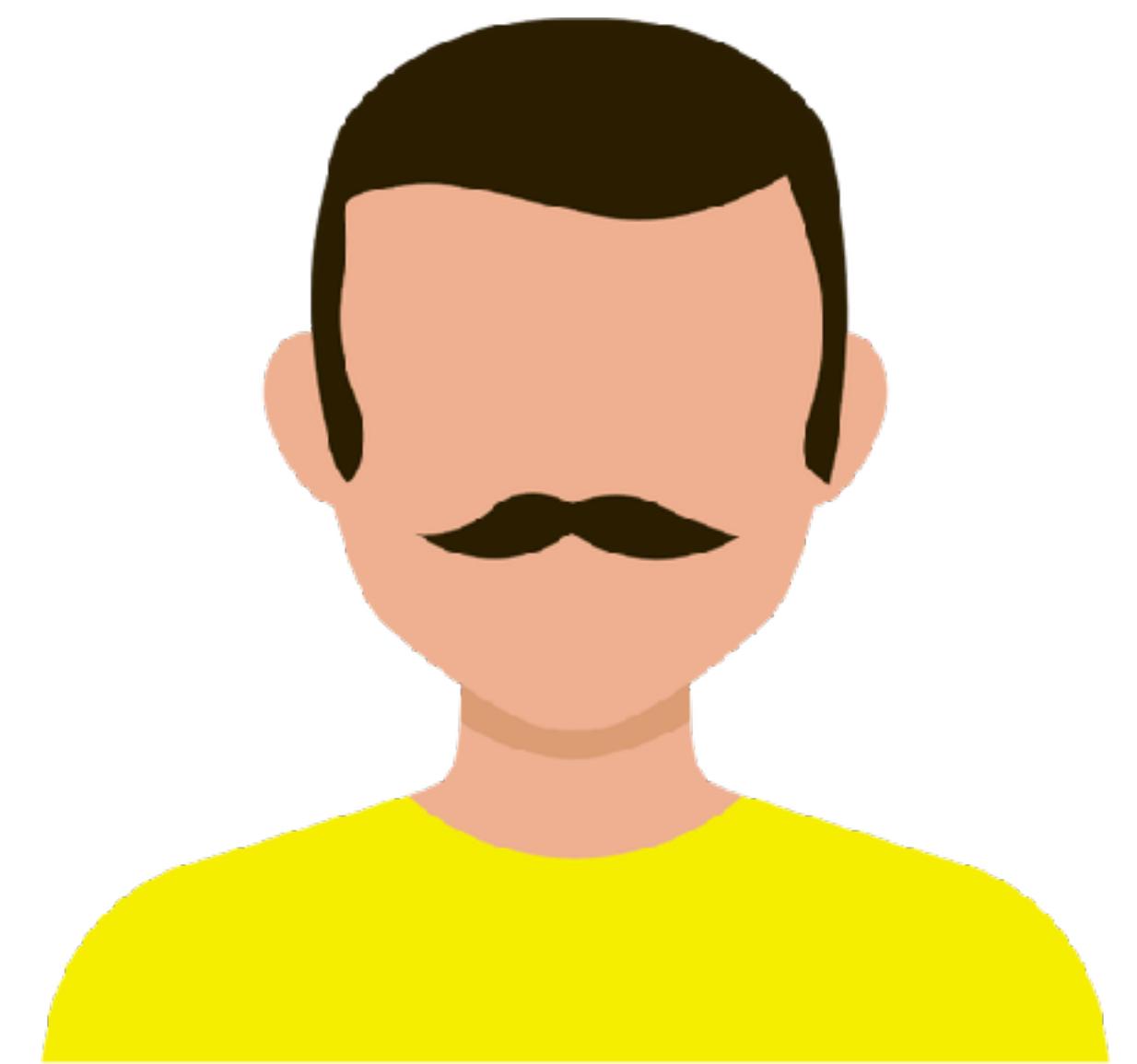
No
mental model

Competent
Practitioner



Useful
mental model

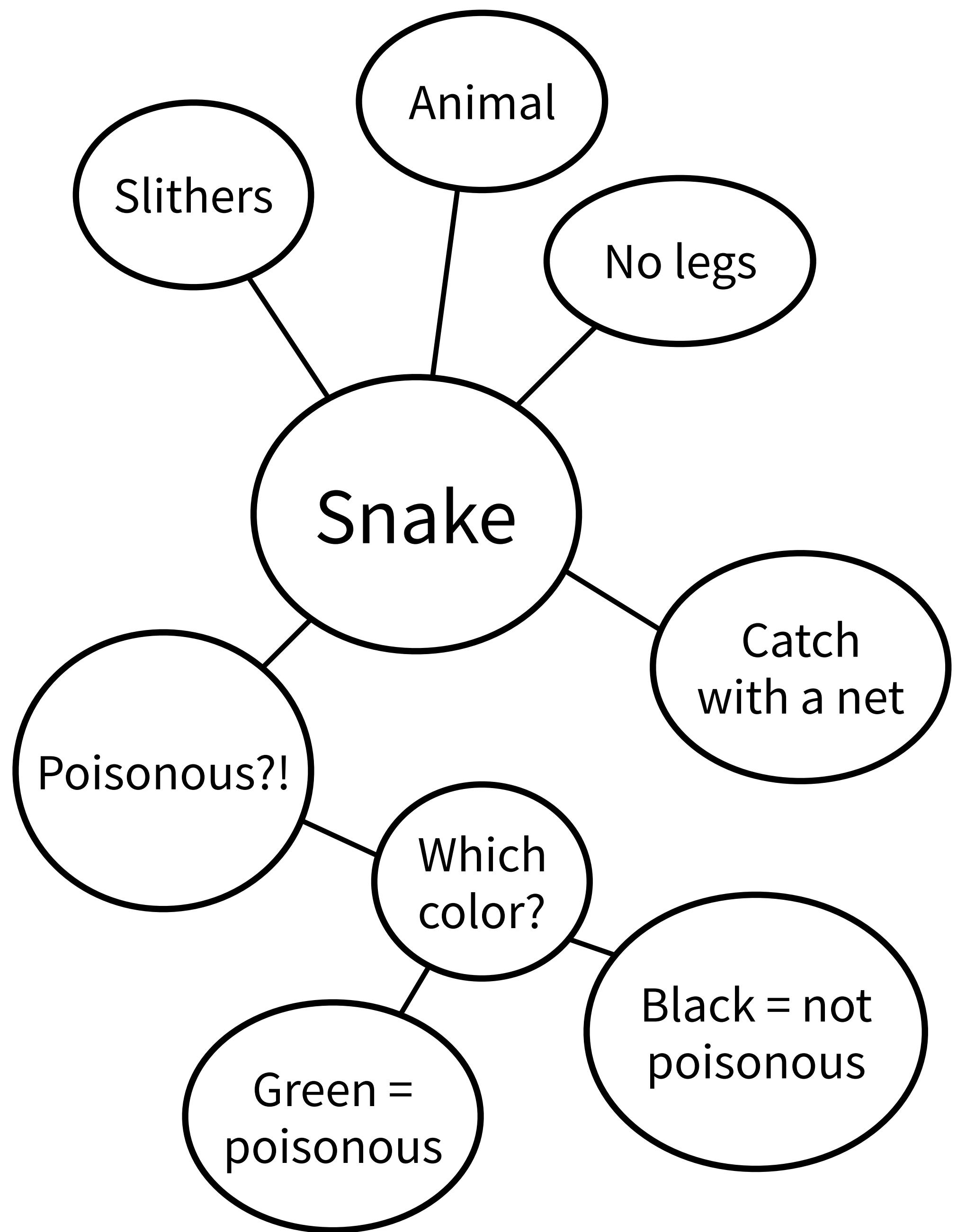
Expert



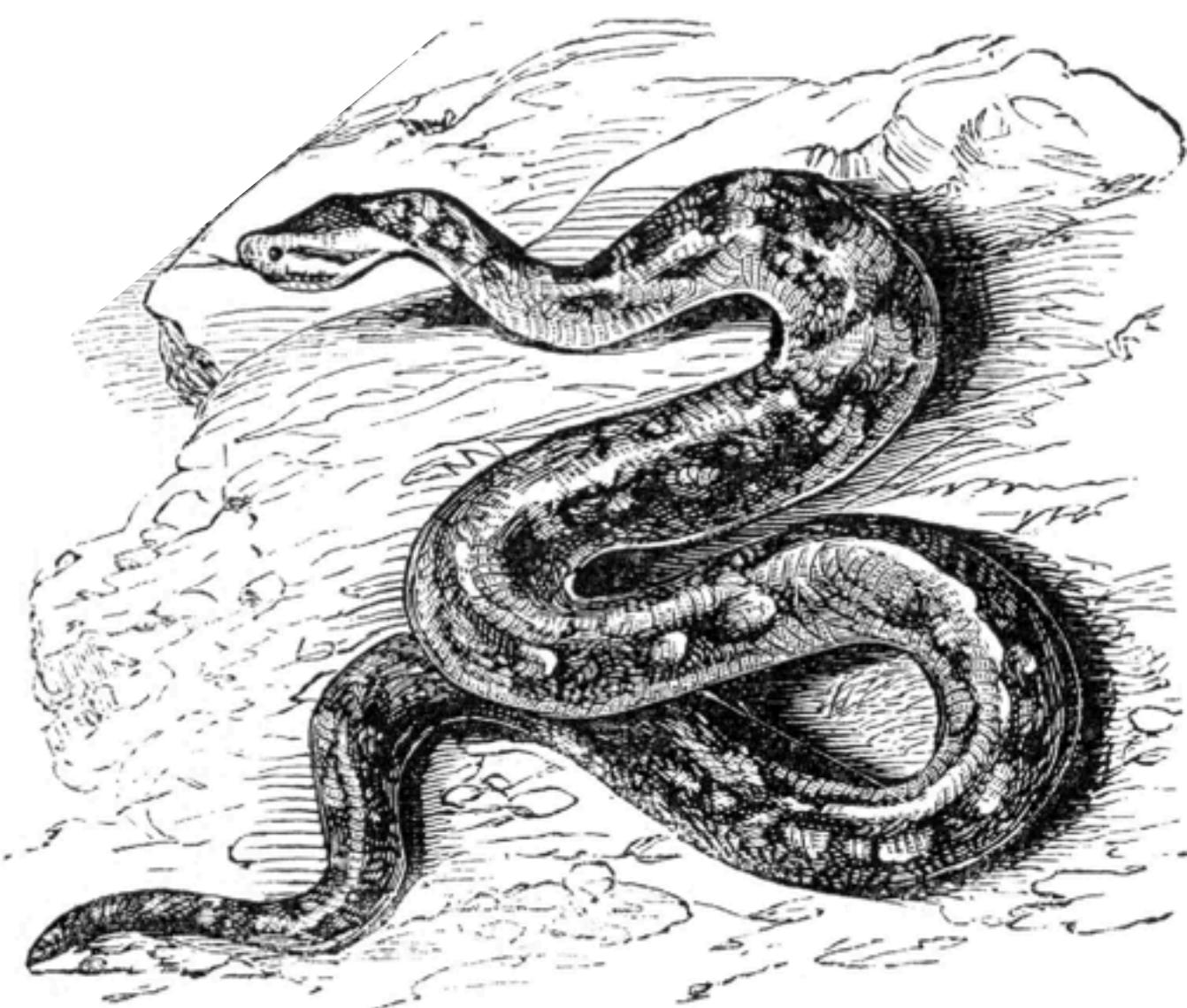
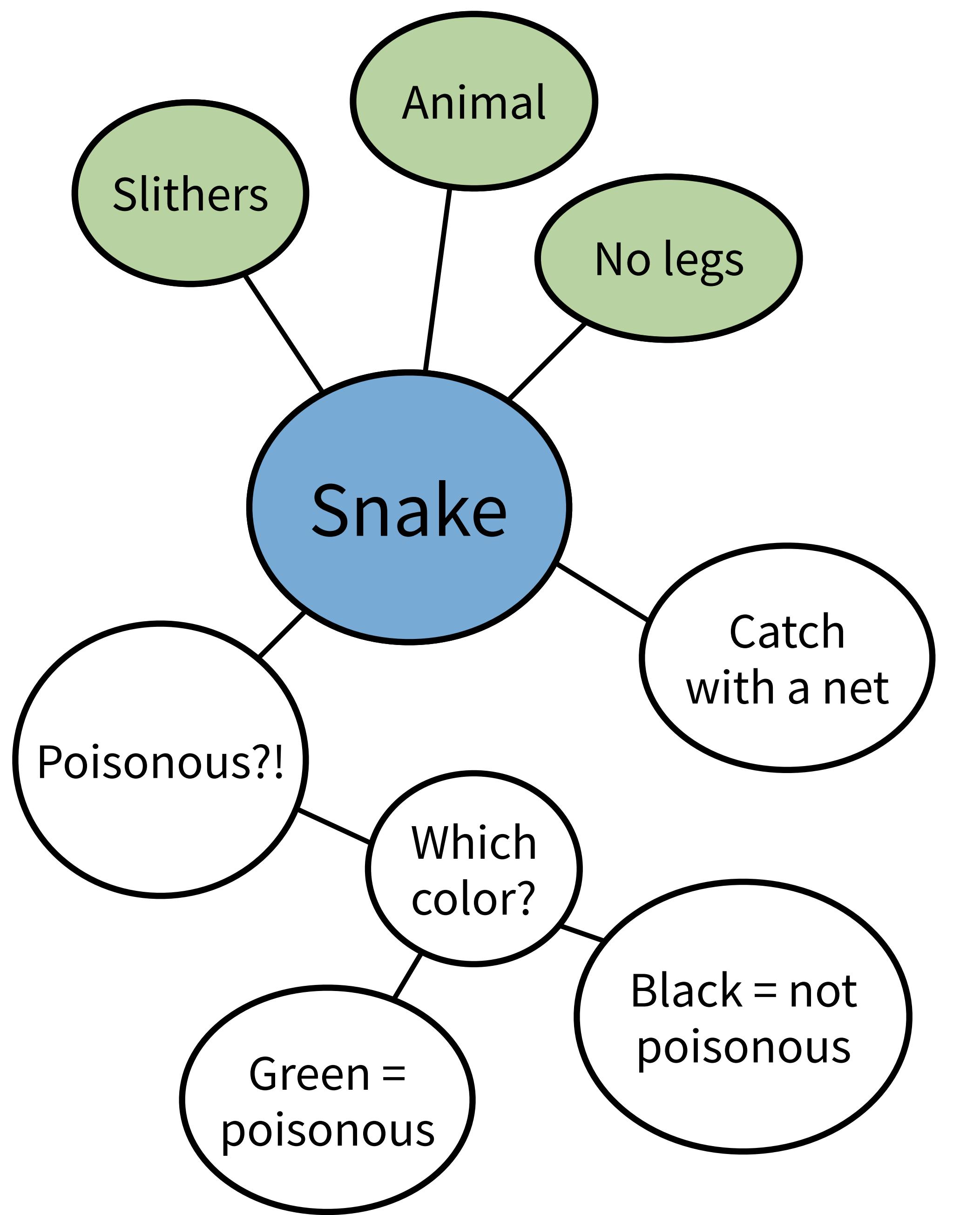
Elaborate
mental models

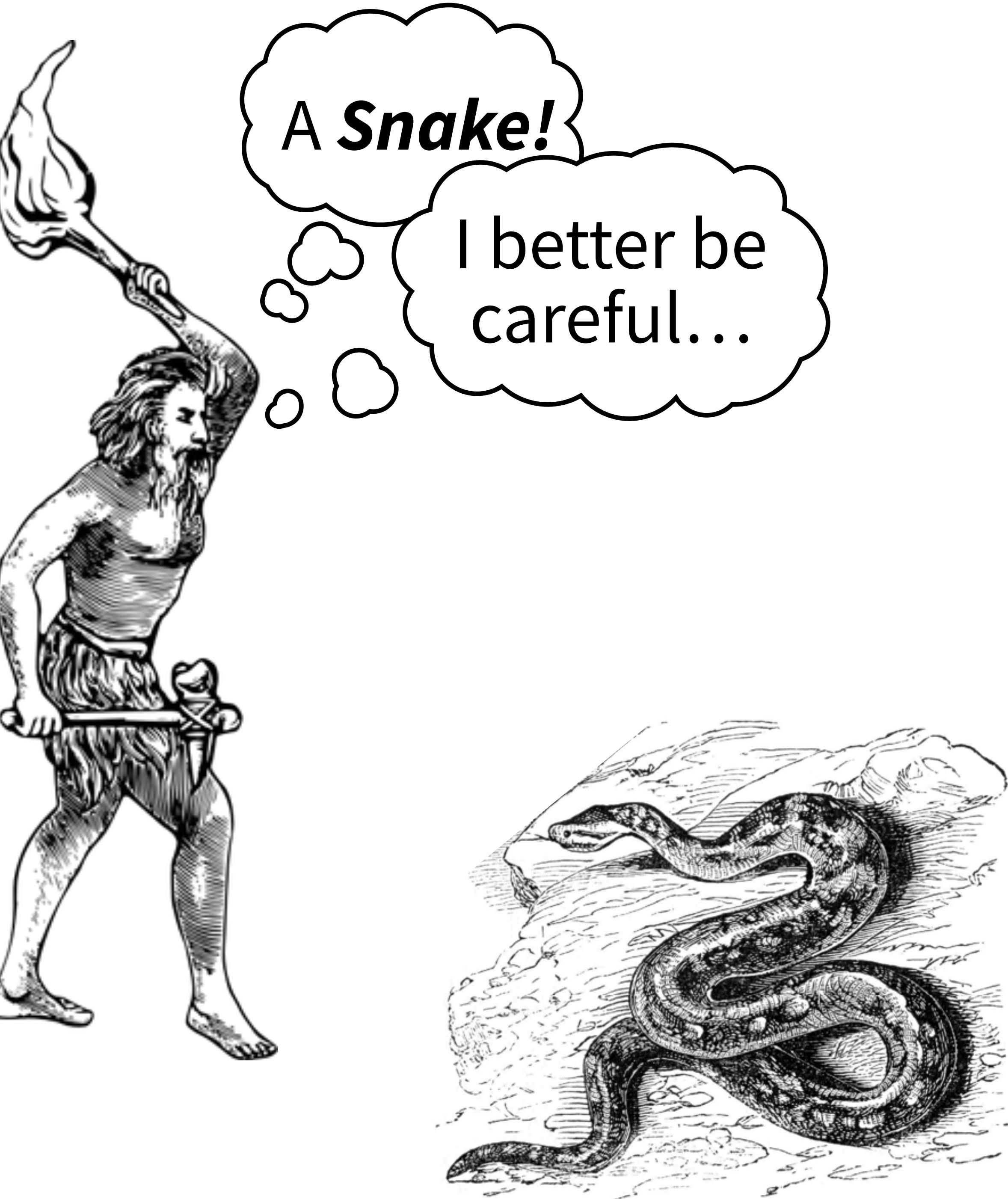
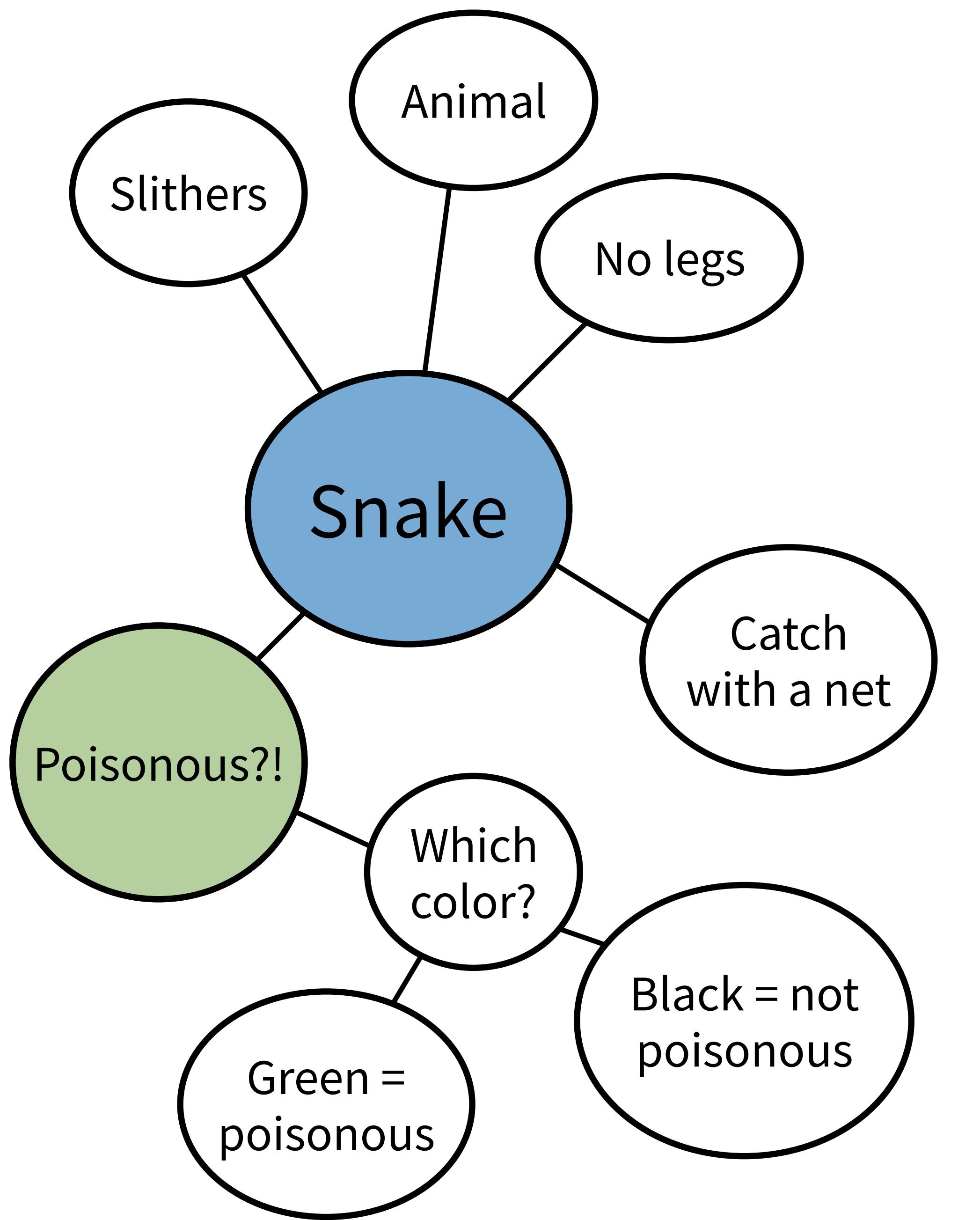
Mental Models

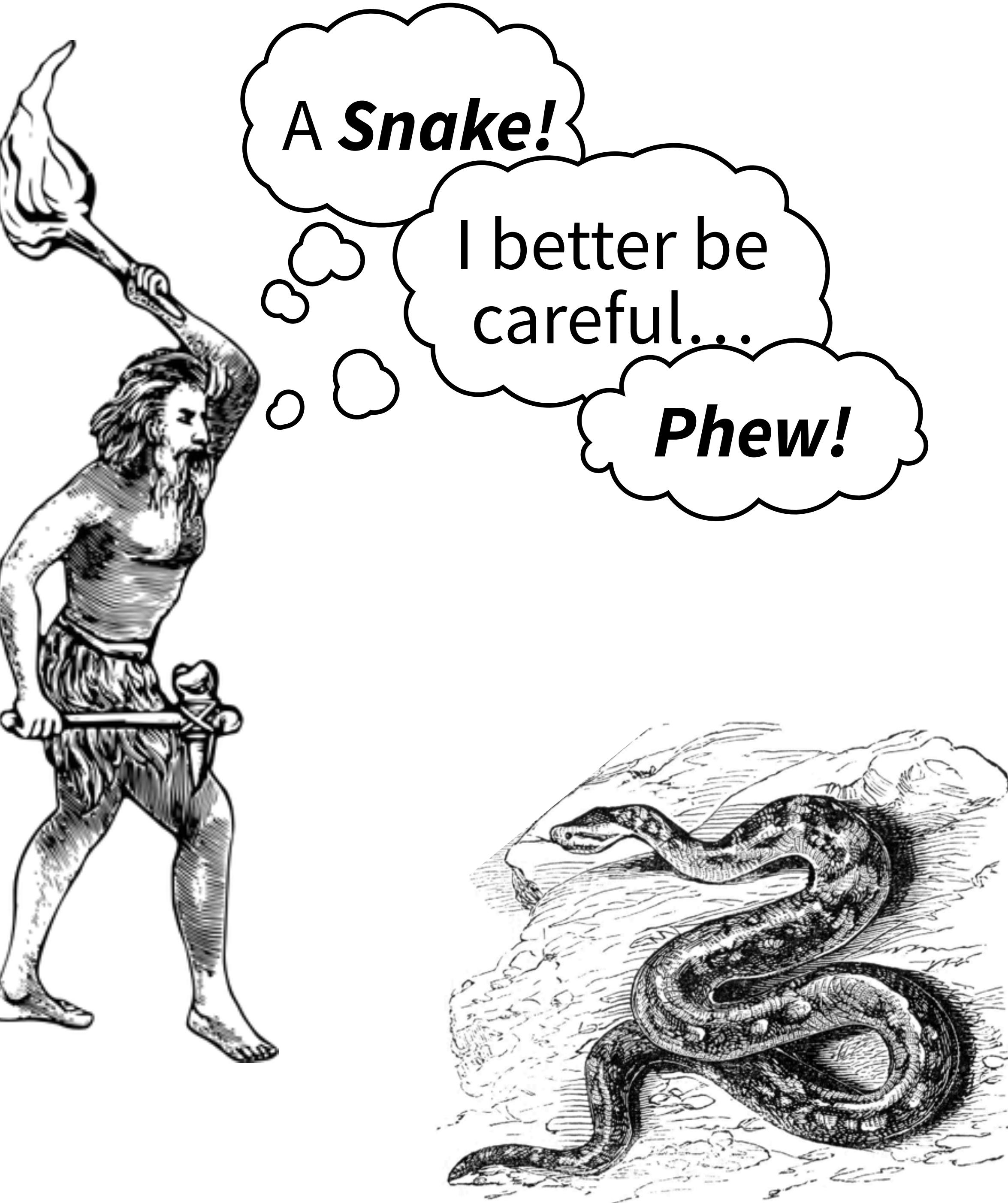
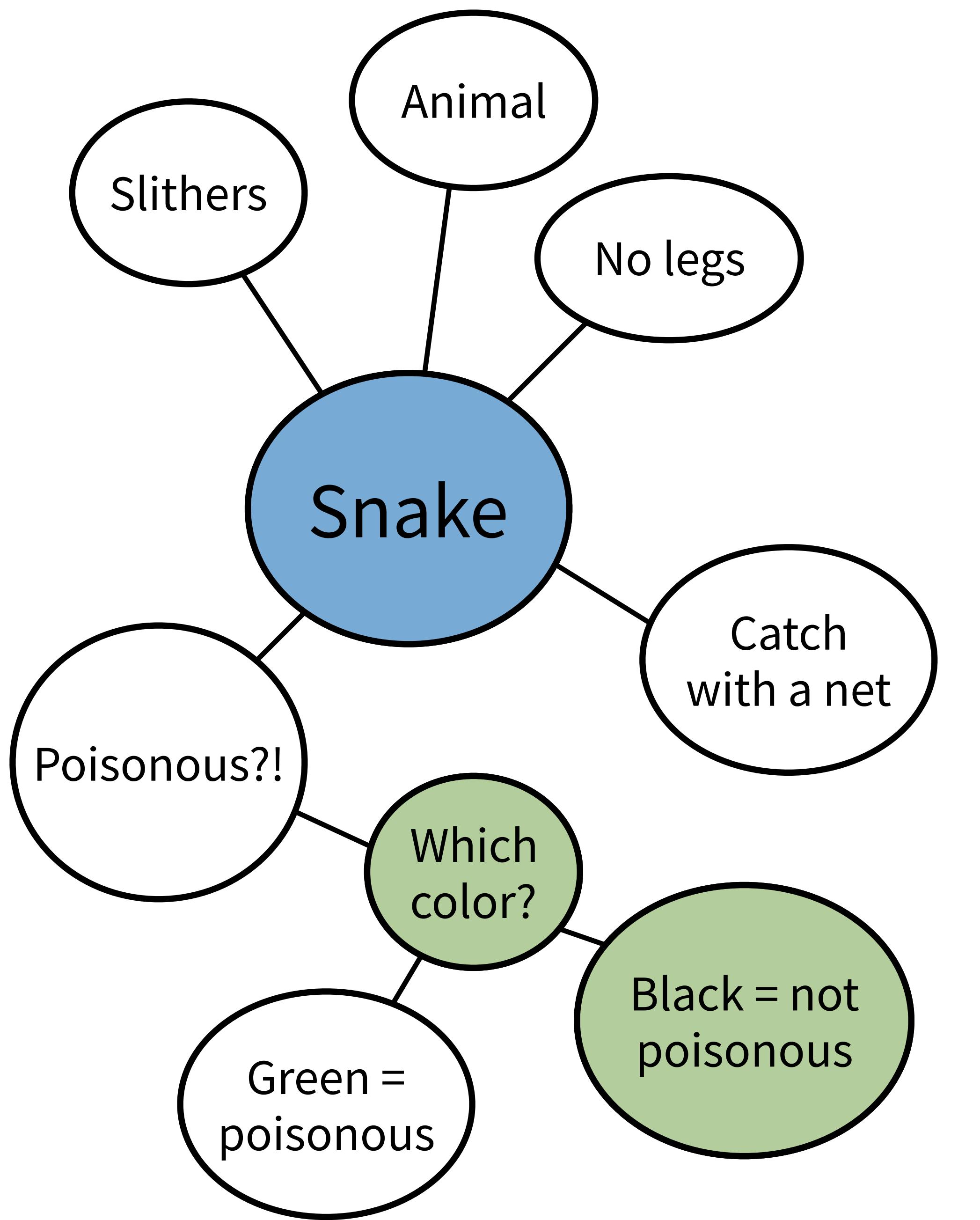


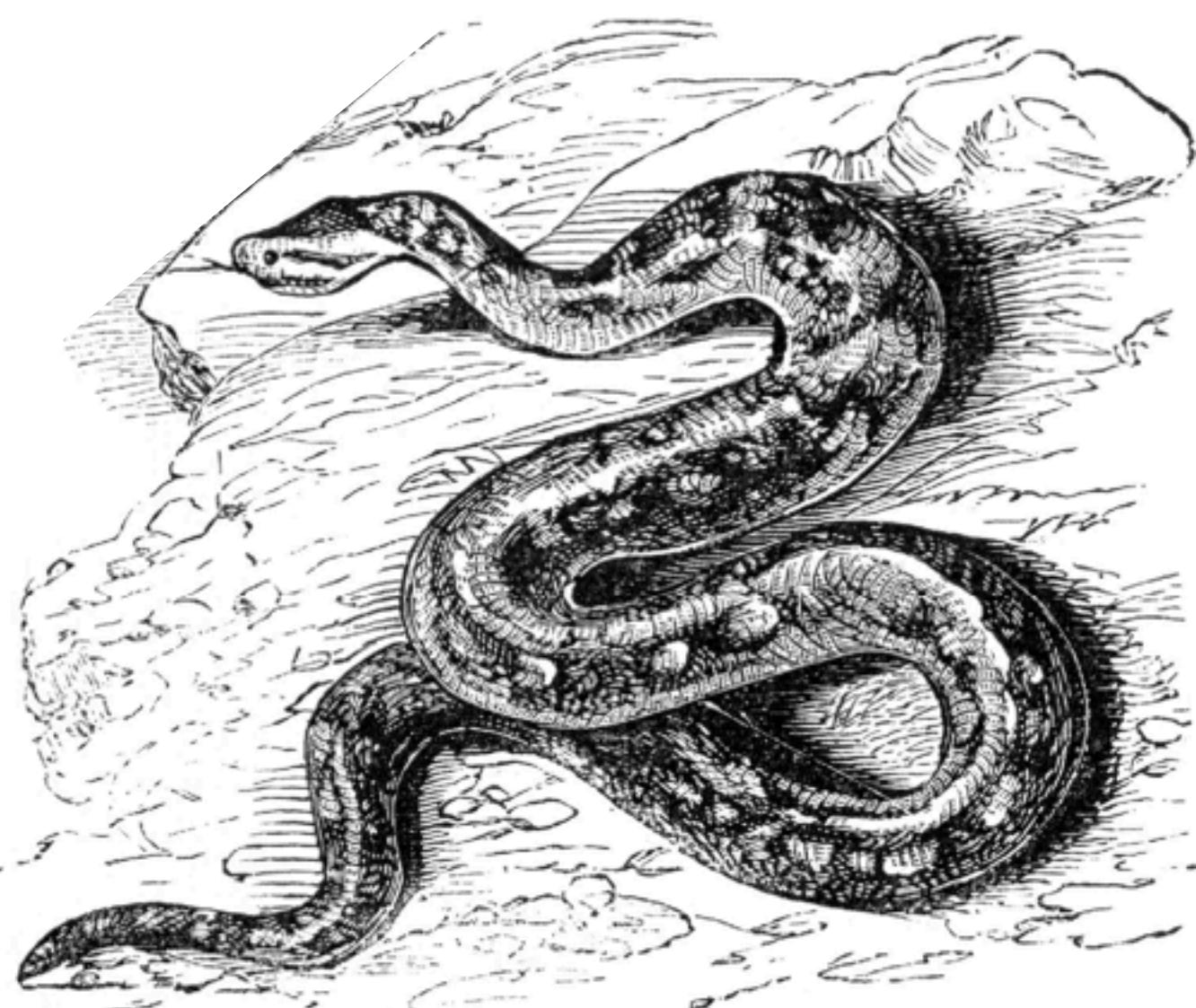
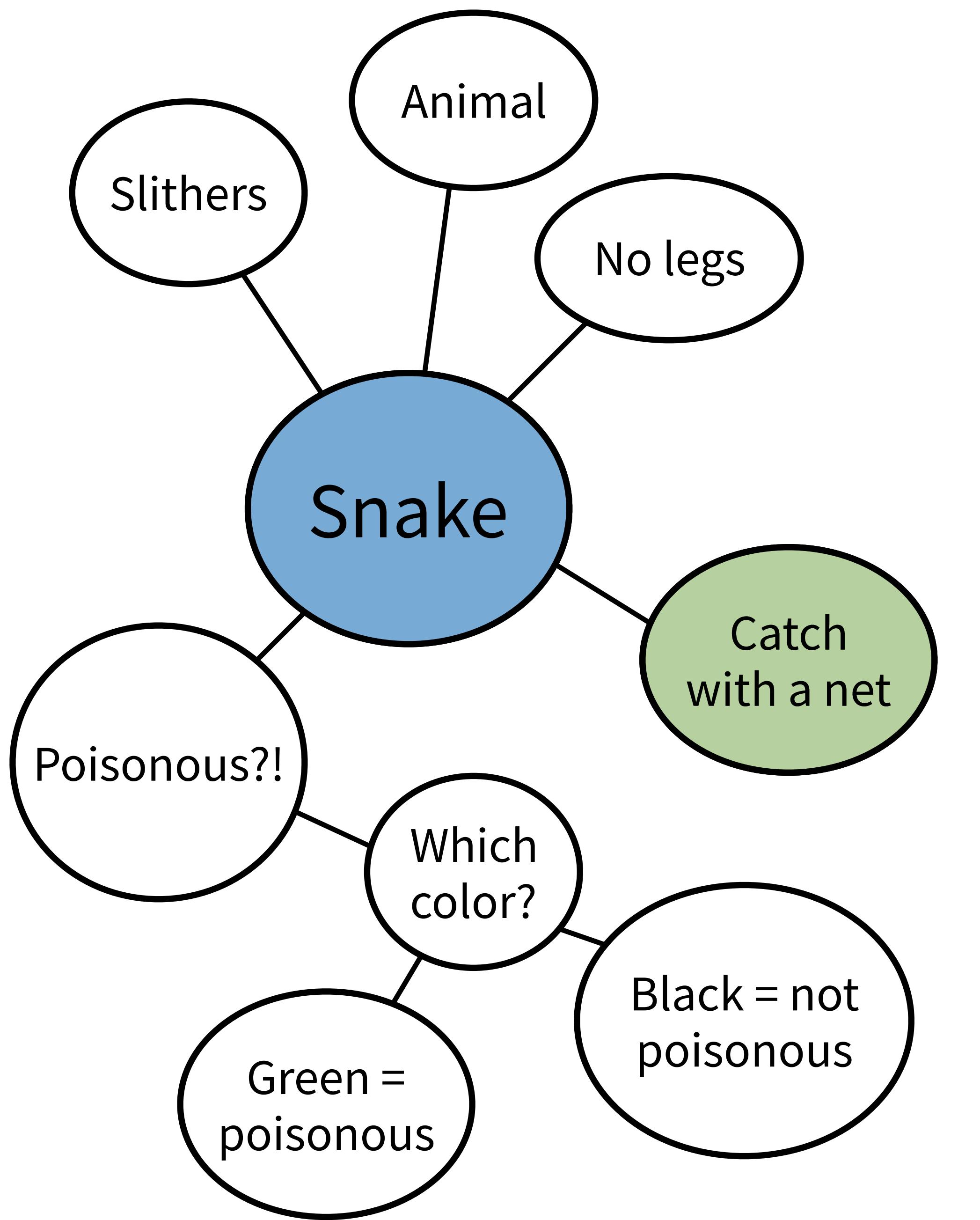


Mental Model
a structure that organizes
facts according to their
relationships











Mental Model

a structure that organizes facts according to their relationships

Your Turn

Open the Master the Tidyverse directory at

github.com/rstudio-education/master-the-tidyverse

Scan the file names and contents, then jot down a quick
mental model of the course.



0. Preclass loop

00. Introduction

- R Notebook (exercises)
- pdf (slides)

00. Reintroduction

- pdf (slides)

1. Visualize Data

- R Notebook (exercises)
- pdf (slides)

2. Transform Data

- R Notebook (exercises)
- pdf (slides)

3. Tidy Data

- R Notebook (exercises)
- pdf (slides)

4. Import Data

- R Notebook (exercises)
- pdf (slides)

5. Data Types

- R Notebook (exercises)
- pdf (slides)

6. Iteration

- R Notebook (exercises)
- pdf (slides)

7. Models

- R Notebook (exercises)
- pdf (slides)

8. List columns

- R Notebook (exercises)
- pdf (slides)

9. Miscellaneous files

- License
- README
- nimbus.csv (a data set)

Beginner



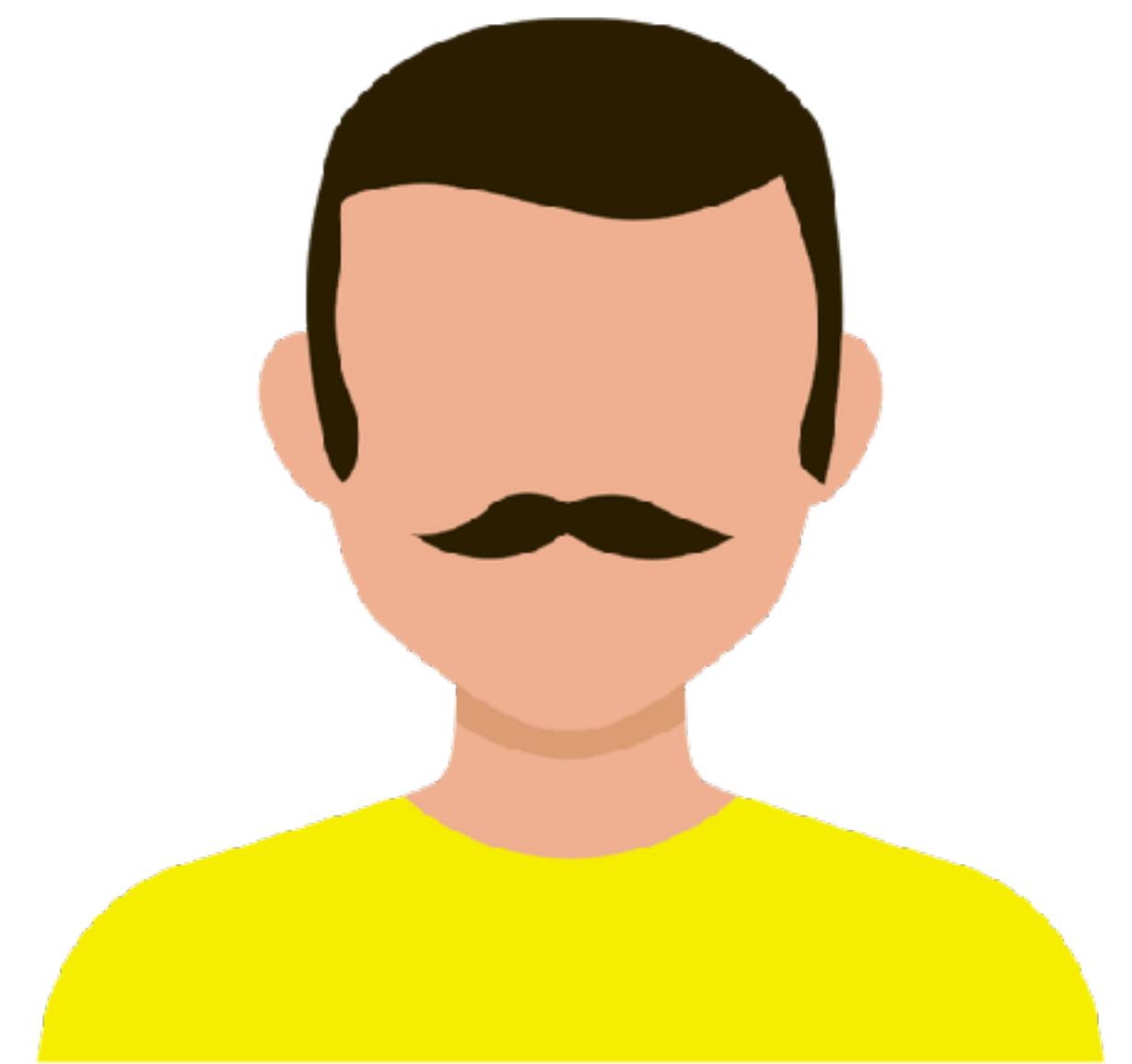
No
mental model

Competent
Practitioner



Useful
mental model

Expert



Elaborate
mental models

Beginner



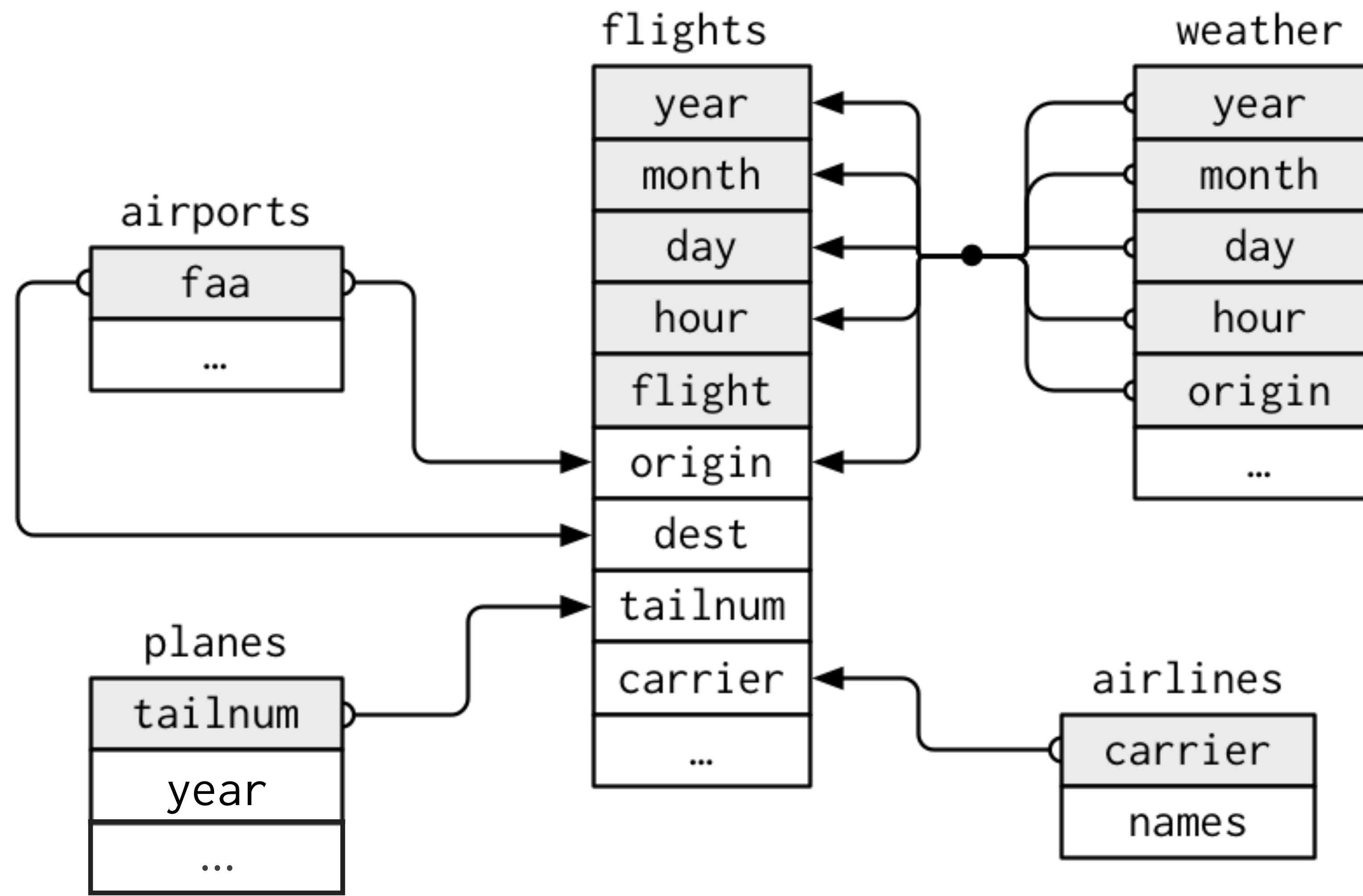
No
mental model

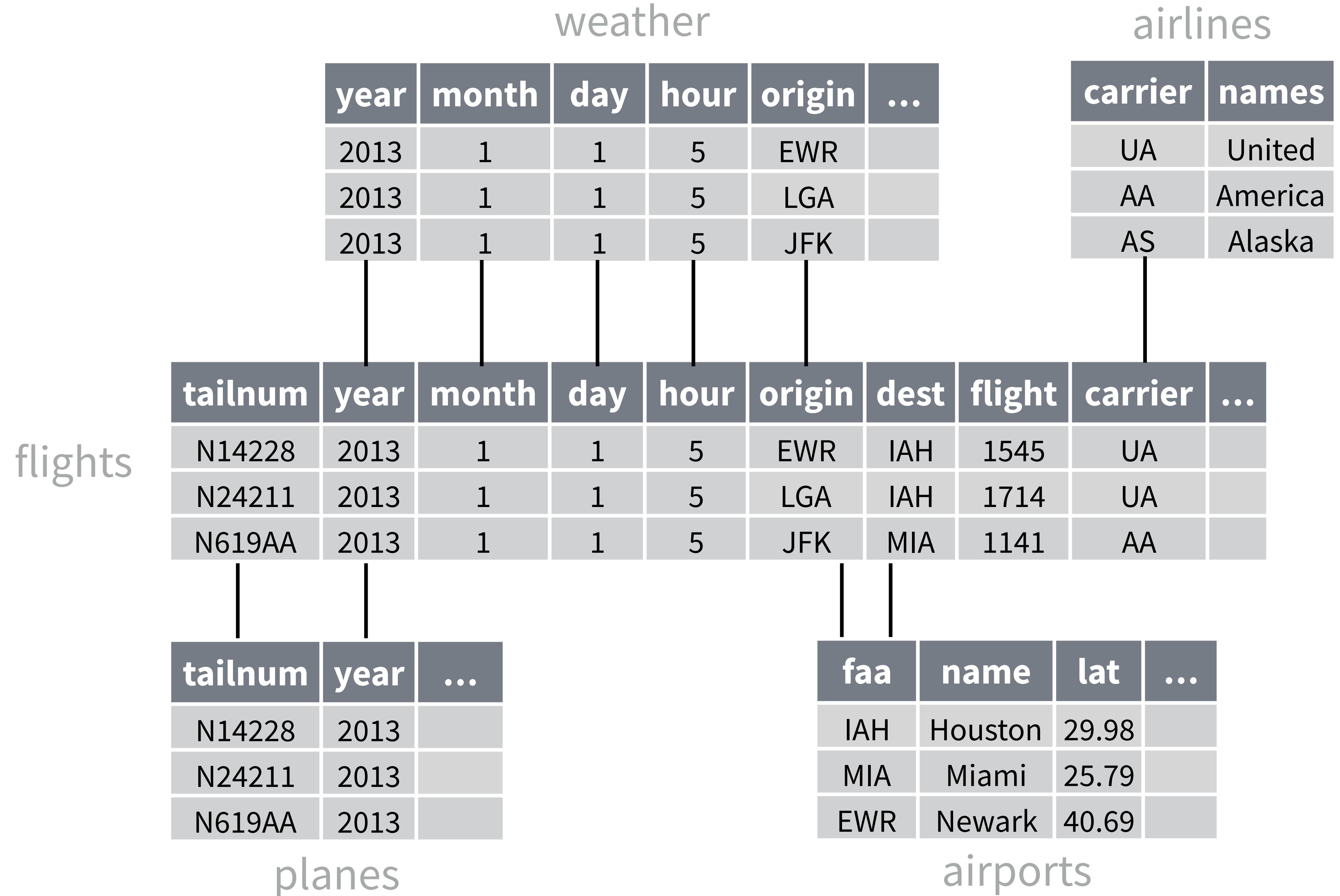
Competent
Practitioner

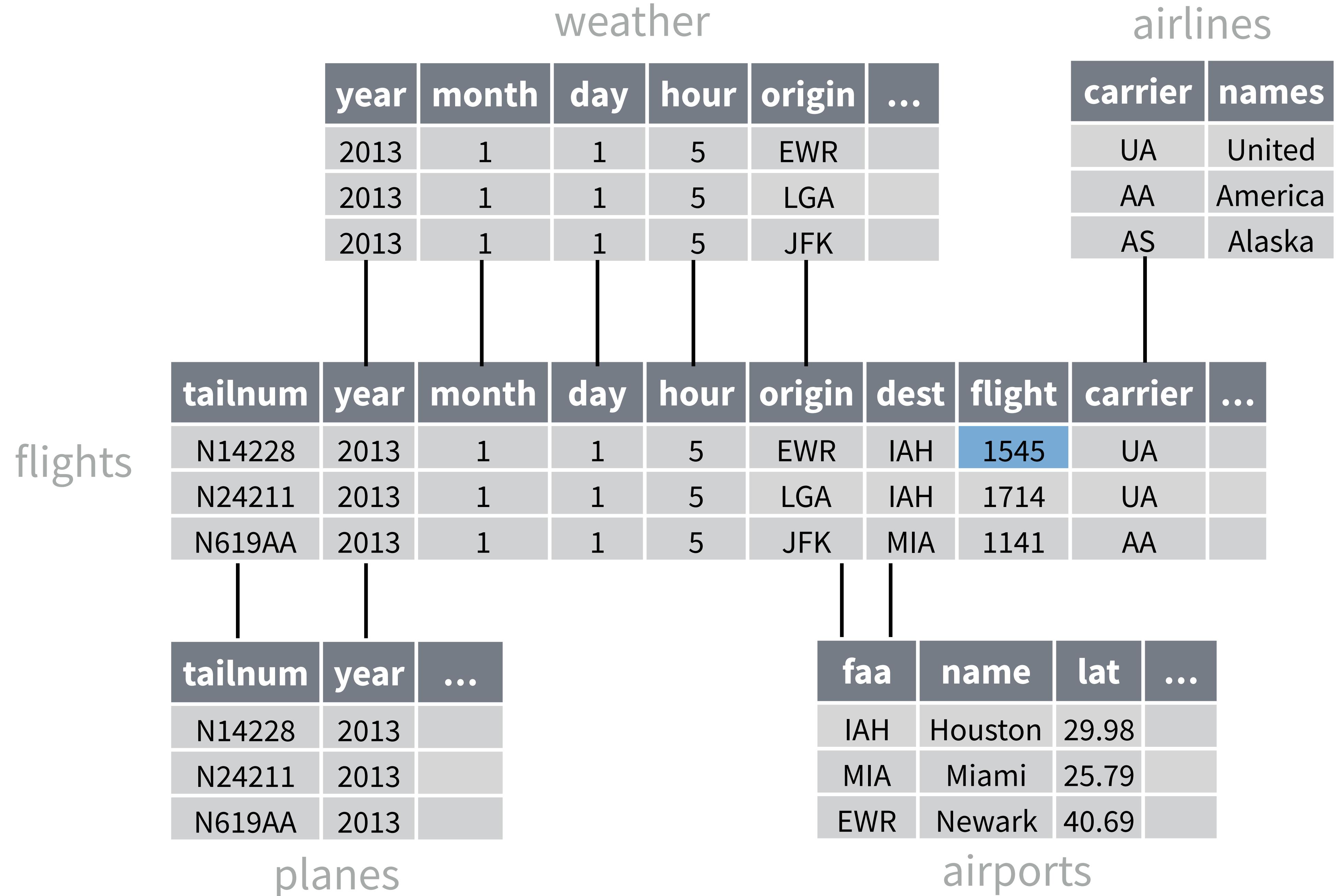


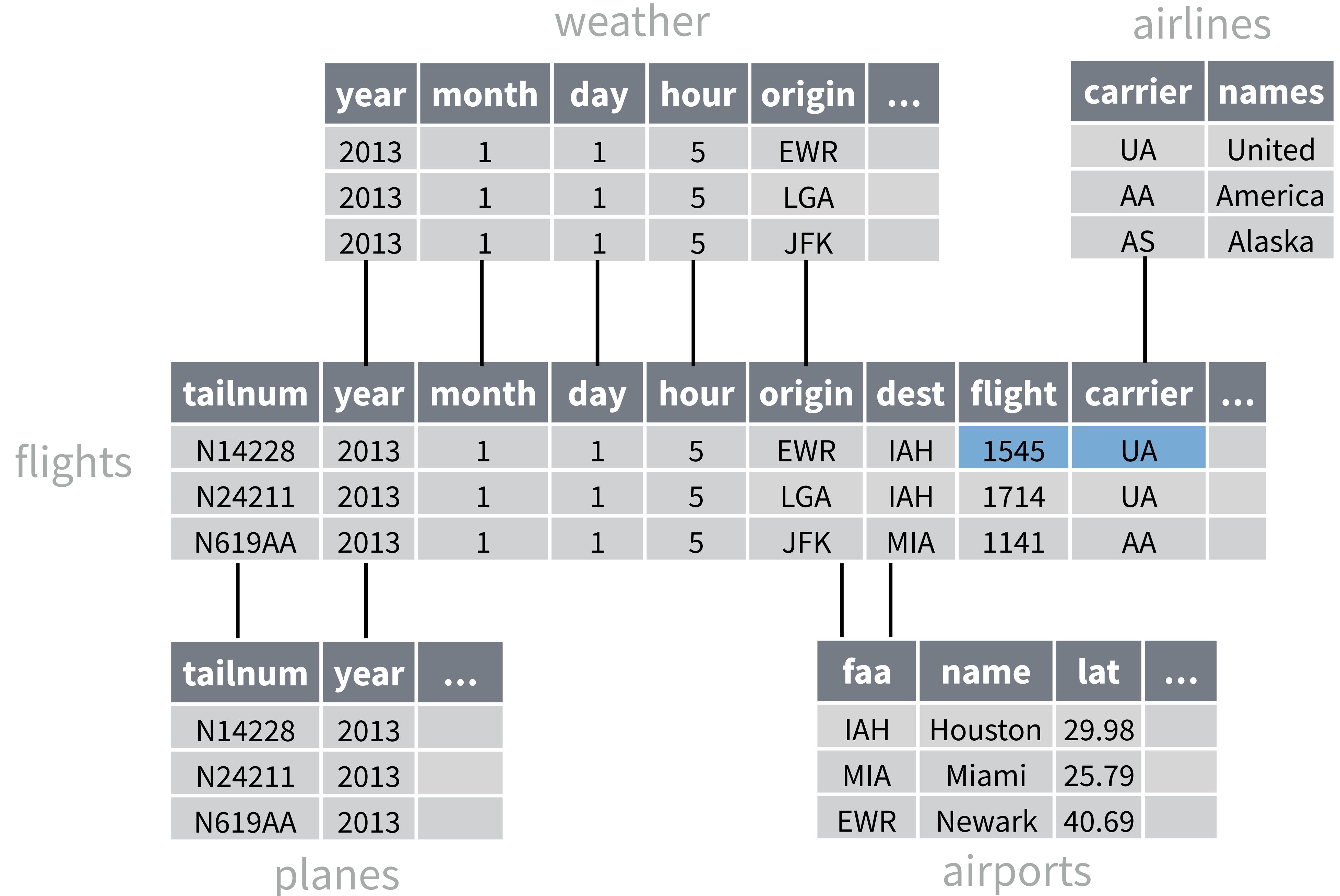
Useful
mental model

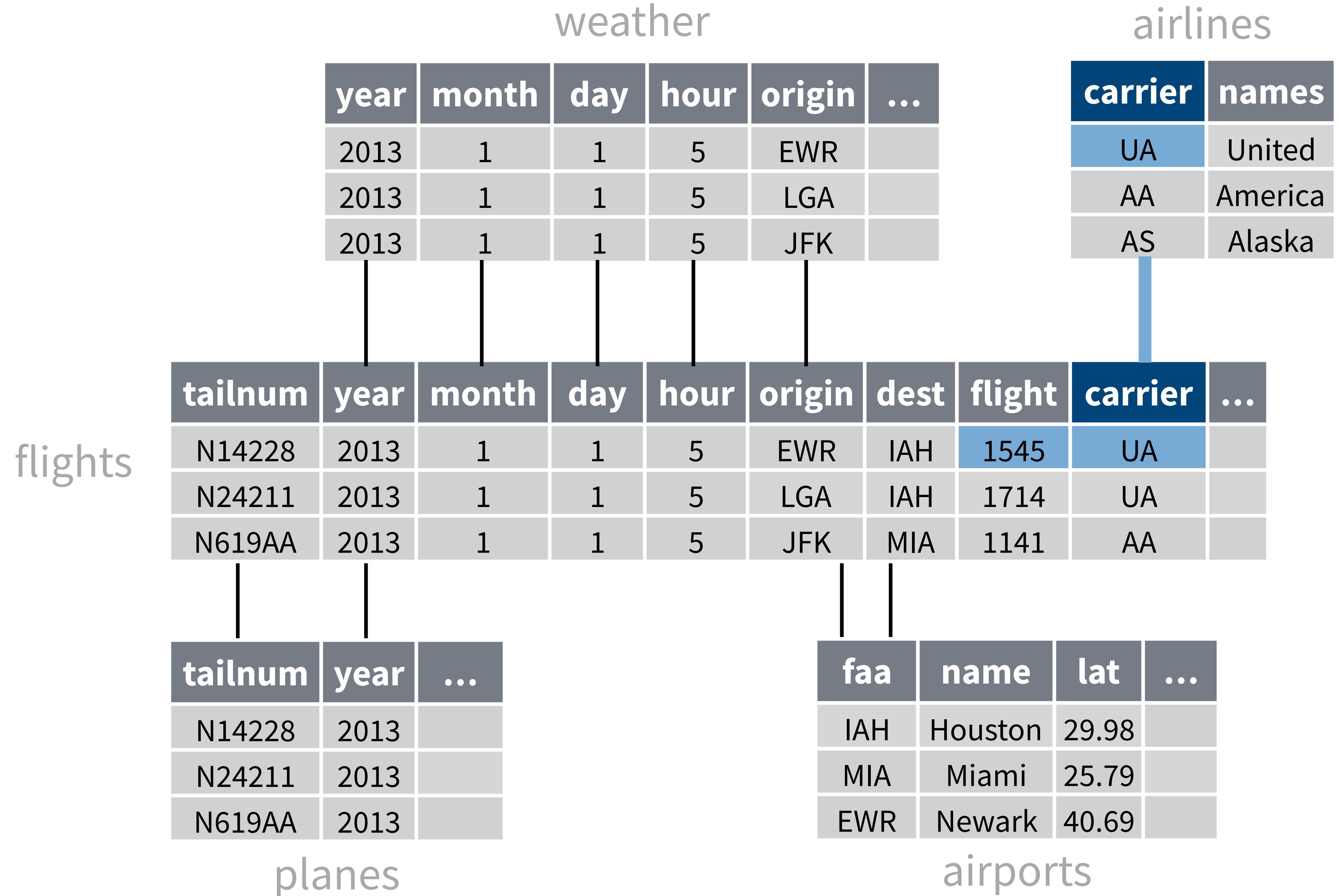
nycflights13

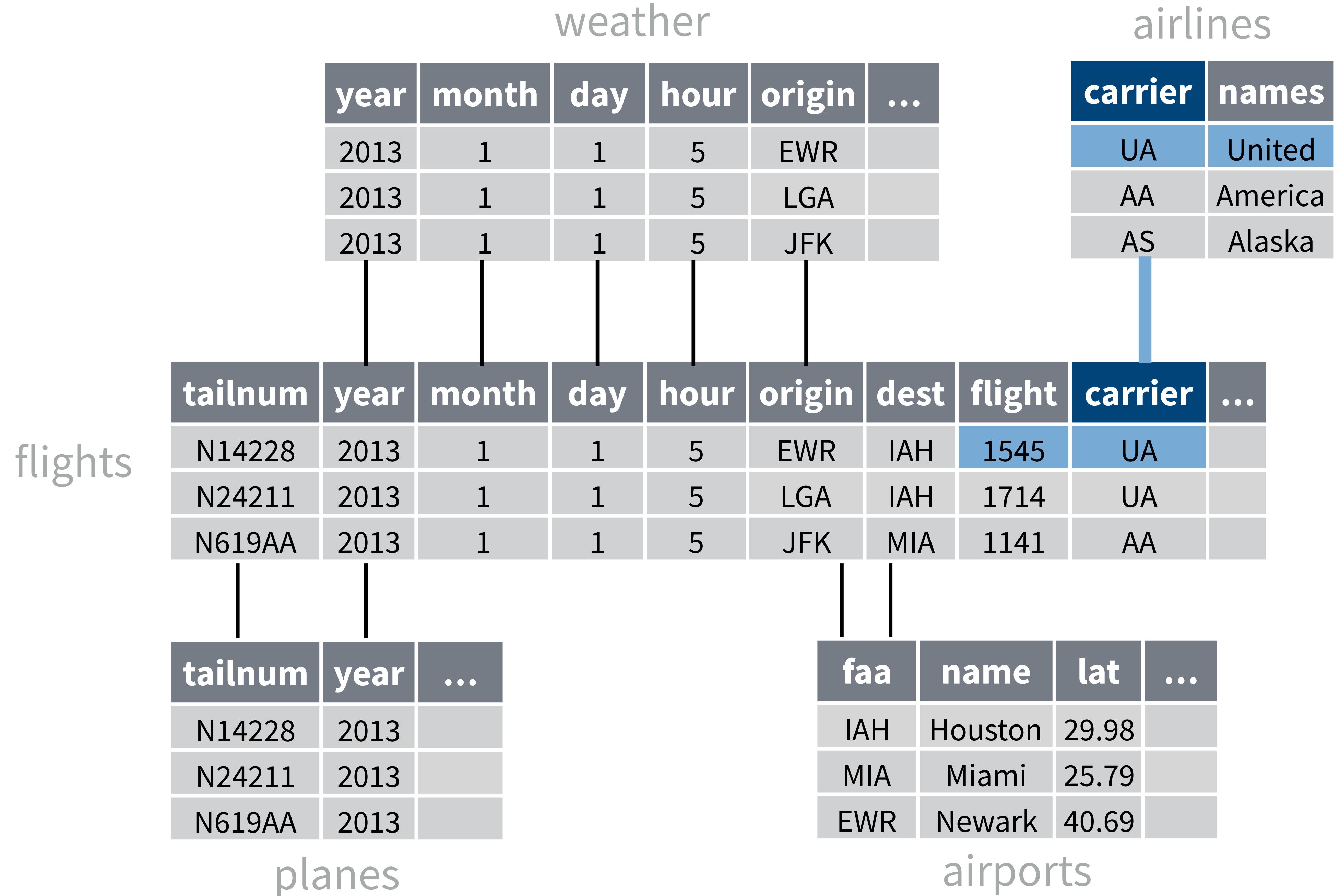


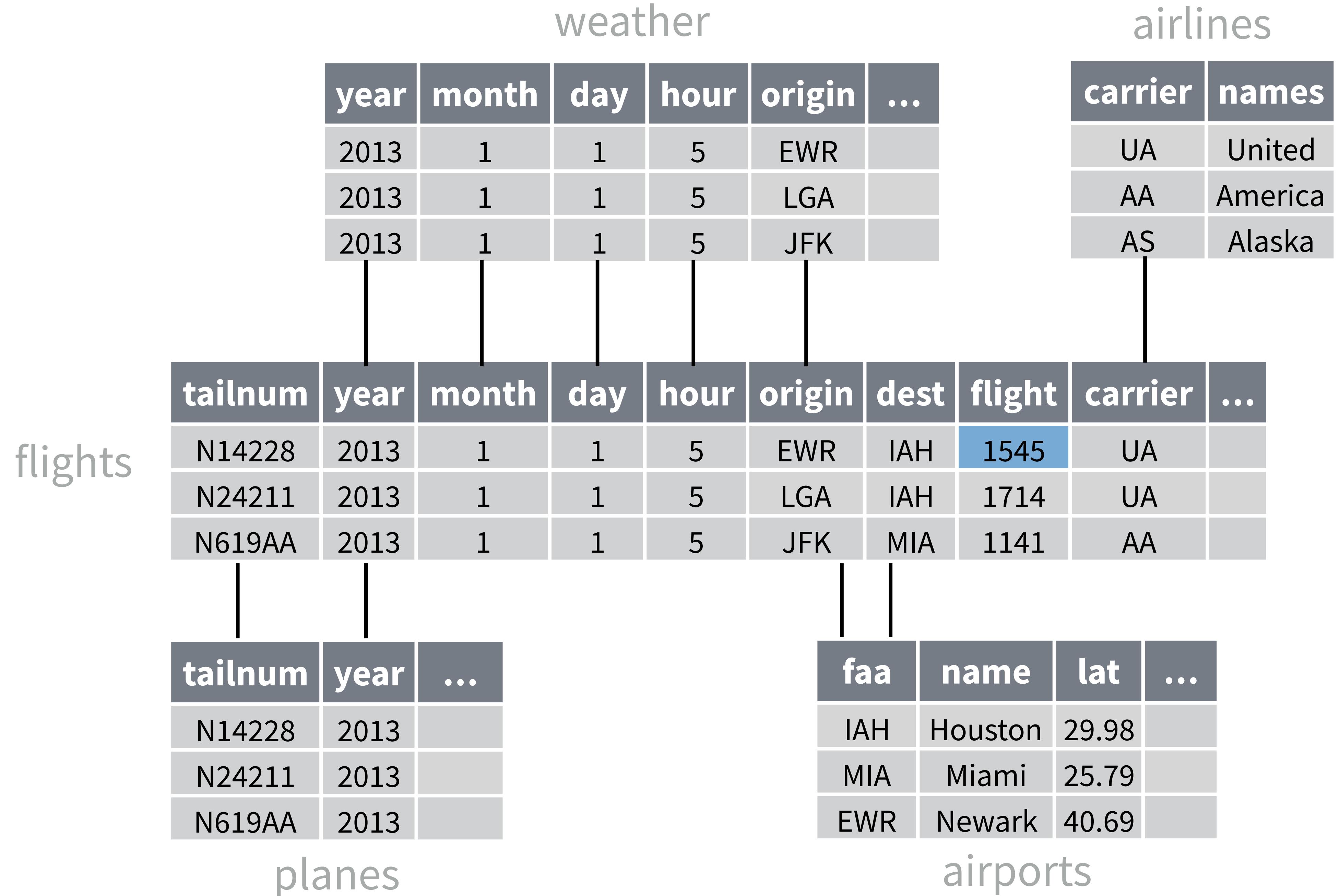


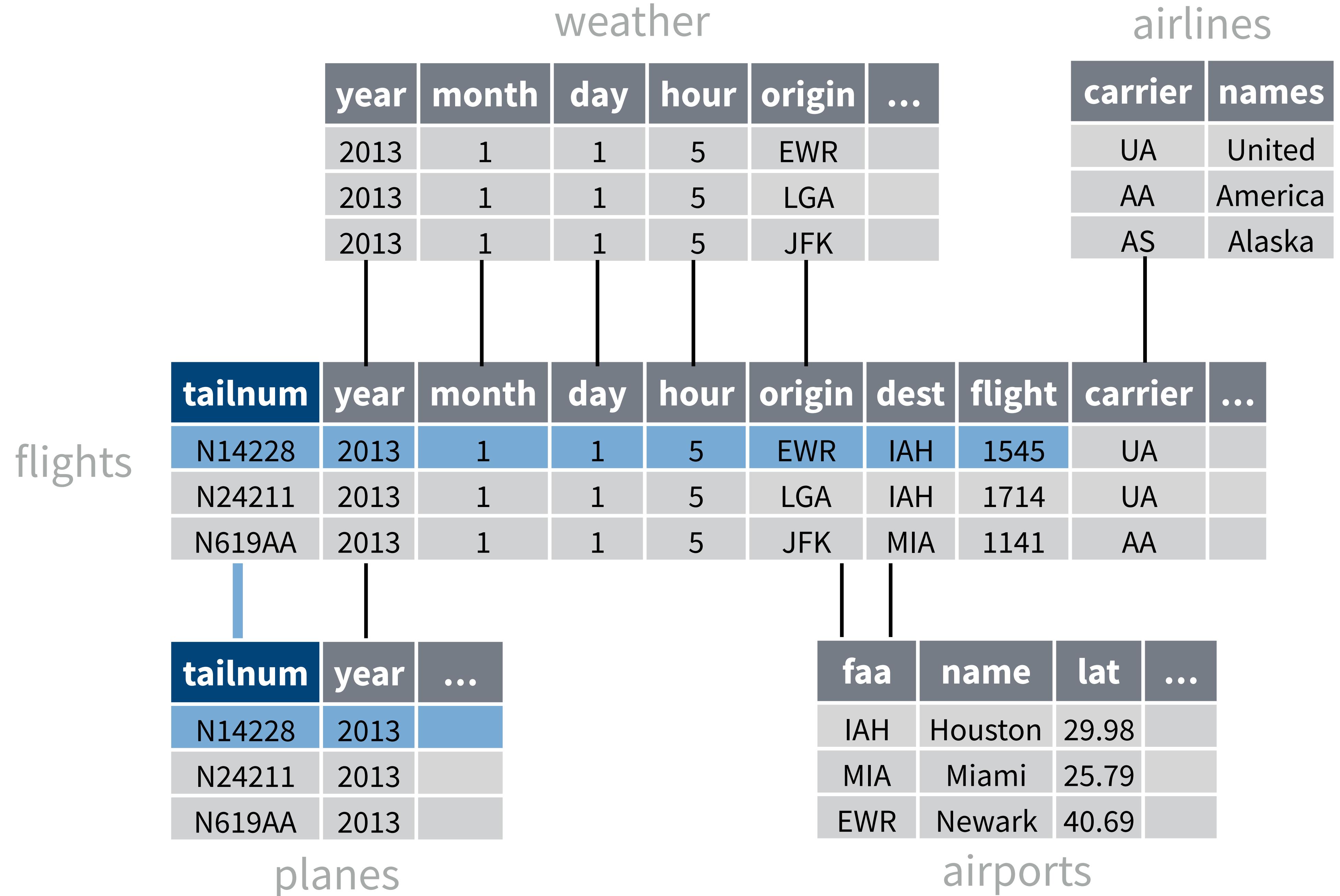


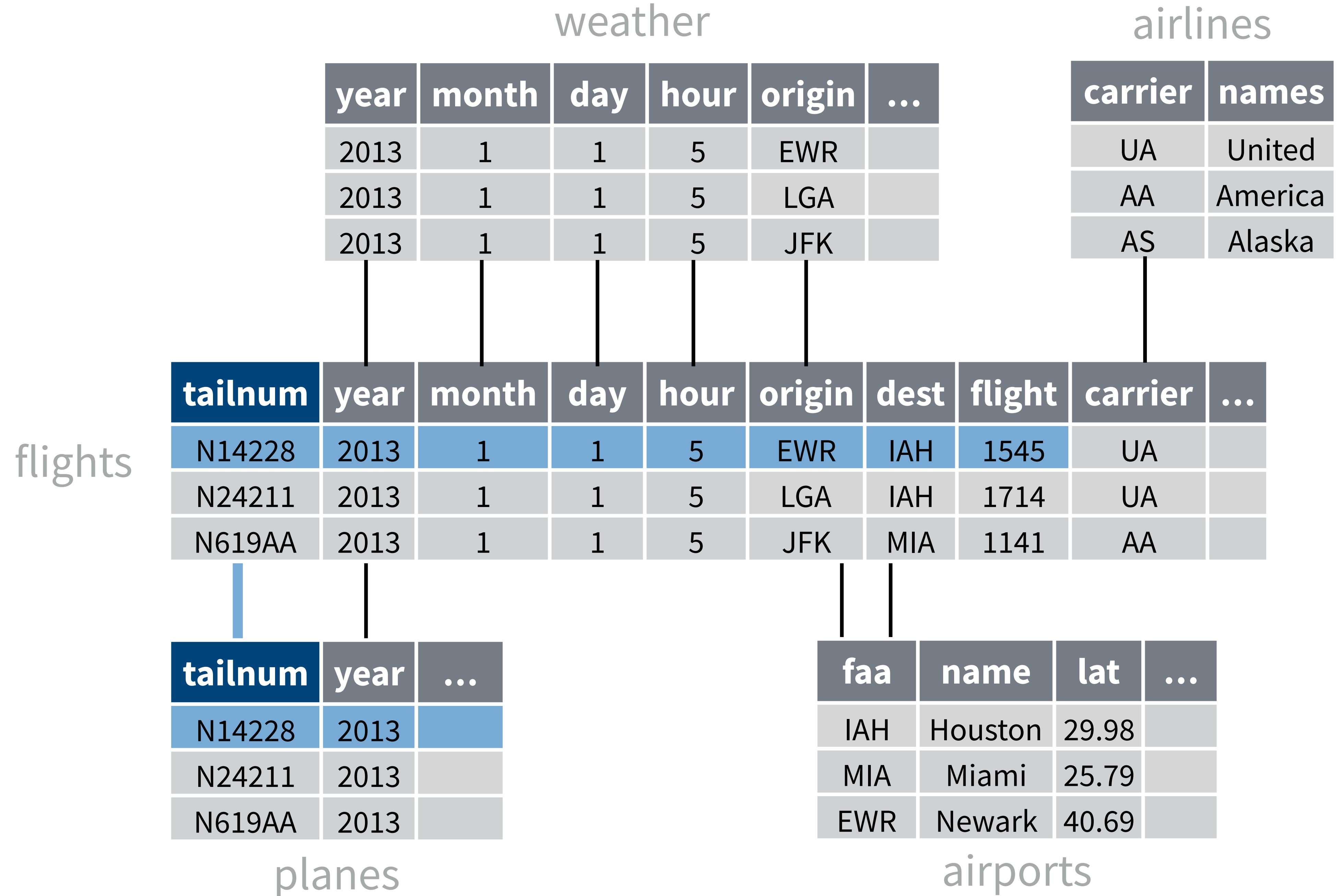




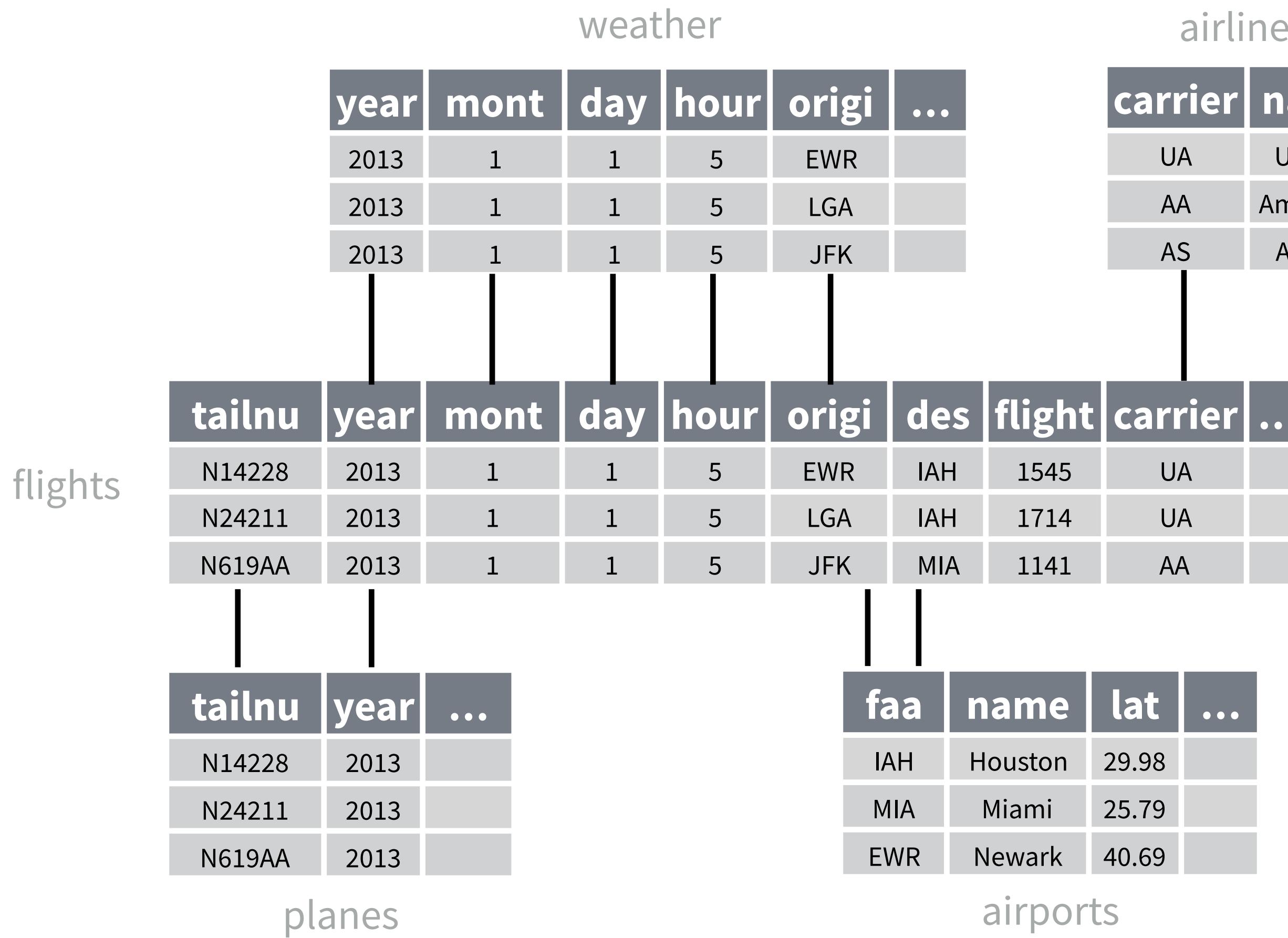








Competent Practitioner



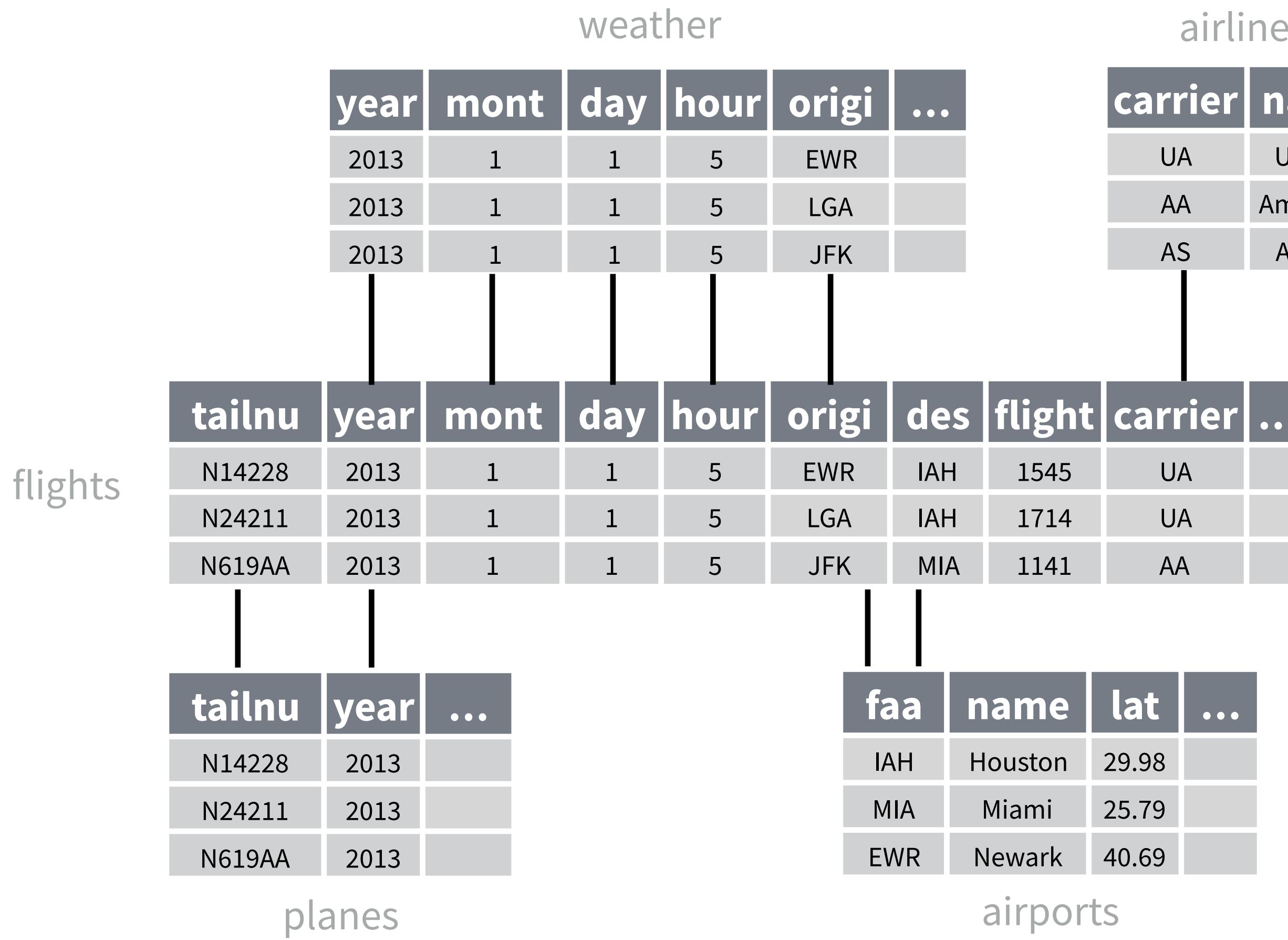
Useful mental model

Your Turn

Consider your area of expertise. Describe the mental model that you use to frame and understand your work.



Competent Practitioner



Useful
mental model

flights

weather

airlines

year	month	day	hour	origin	...
2013	1	1	5	EWR	
2013	1	1	5	LGA	
2013	1	1	5	JFK	

carrier	names
UA	United
AA	America
AS	Alaska

tailnum	year	month	day	hour	origin	dest	flight	carrier	...
N14228	2013	1	1	5	EWR	IAH	1545	UA	
N24211	2013	1	1	5	LGA	IAH	1714	UA	
N619AA	2013	1	1	5	JFK	MIA	1141	AA	

tailnum	year	...	faa	name	lat	...
N14228	2013		IAH	Houston	29.98	
N24211	2013		MIA	Miami	25.79	
N619AA	2013		EWR	Newark	40.69	
			LGA	LaGuardi	40.77	

planes

Beginner



No
mental model

- LGA is LaGuardia Airport
- It rained at JFK on 1/3/2013
- UA is United
- N619AA is a Cessna Plane
- Flight 1532 was delayed on 12/31/13
- Flight 1714 flew from LGA to IAH
- Houston airport is at 29.98 latitude
- Miami is MIA

Beginner



No
mental model

- LGA is LaGuardia Airport weather
- It rai

year	mont	day	hour	origi	...	carrier	name
2013						UA	United
- UA is United airlines
- flights

tailnu	year	mont	day	hour	origi	des	flight	carrier	...
							1532	delayed	12/31/13
- Flight 1532 was delayed on 12/31/13 flights
- tailnu year ...

tailnu	year	...	faa	name	lat	...
			AA	American	29.98	latitude
- 4 flew from AA airports
- Houston airport is at 29.98 latitude airports
- Miami is MIA airports

How to teach a workshop

How to teach a workshop

1. Teach the **mental model(s)**

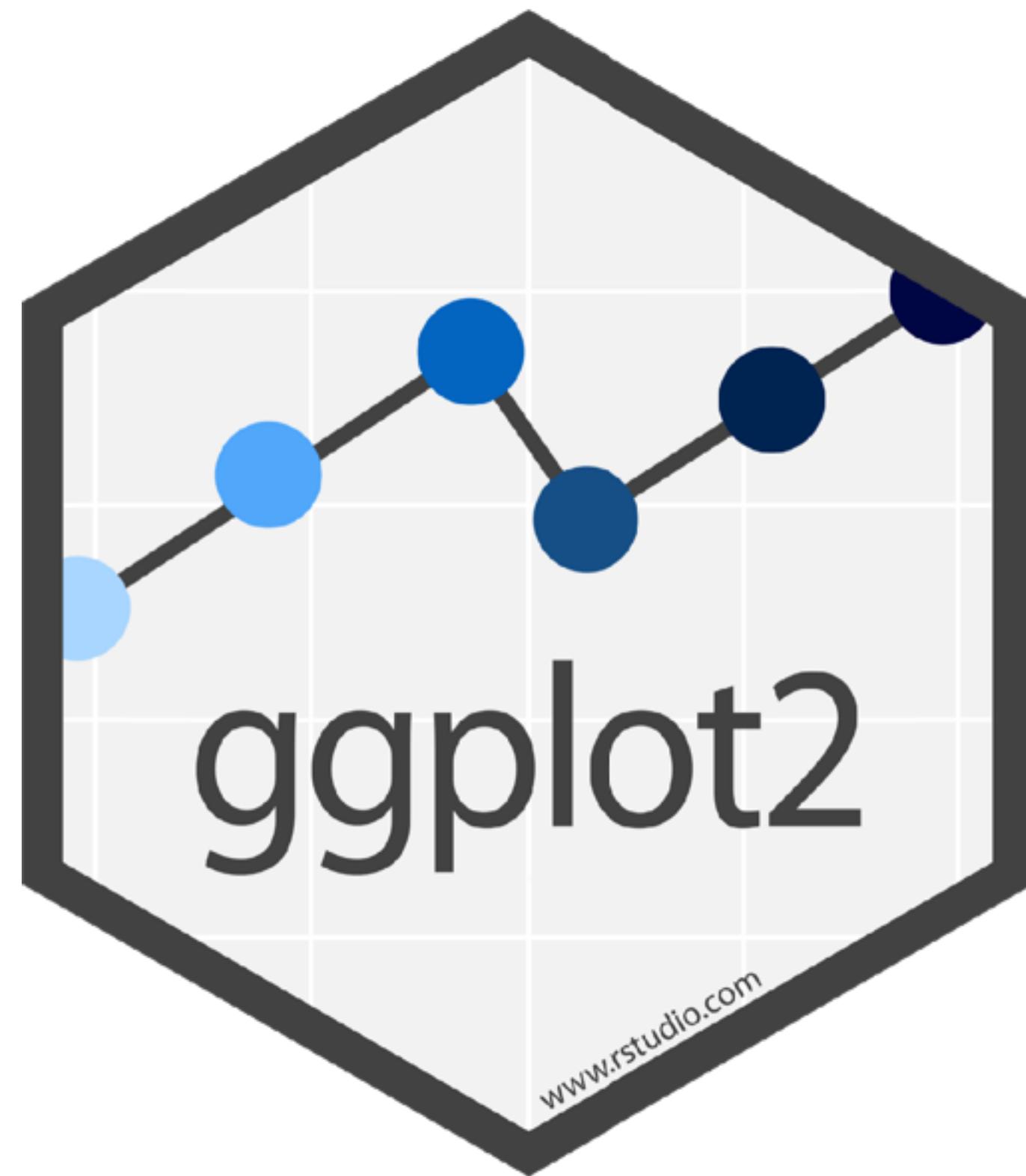
Explain them. Practice the skills they support.

2. Put useful details in a **handout**

3. Follow up with **post-workshop quizzes**

Use cognitive theory as a guide

Visualize Data with



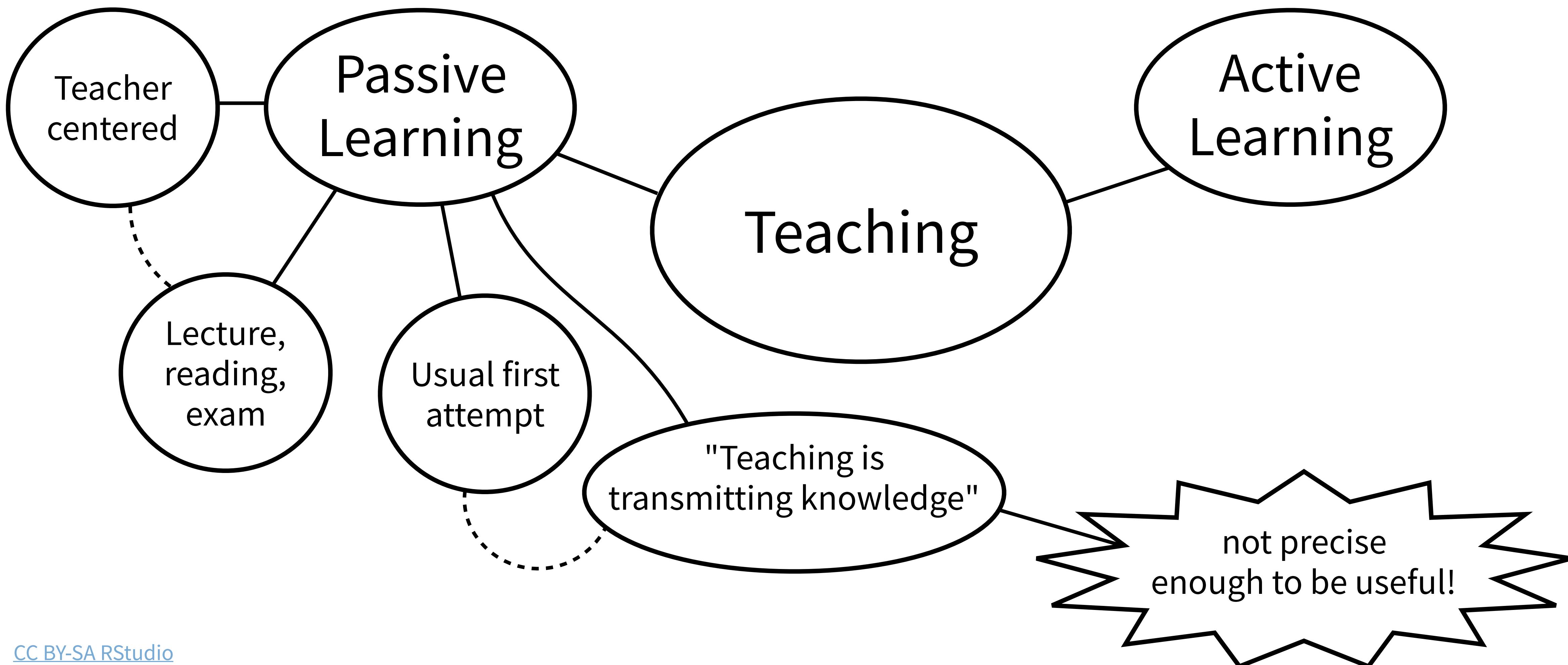
Concept maps



Concept Maps



Concept Maps



Your Turn

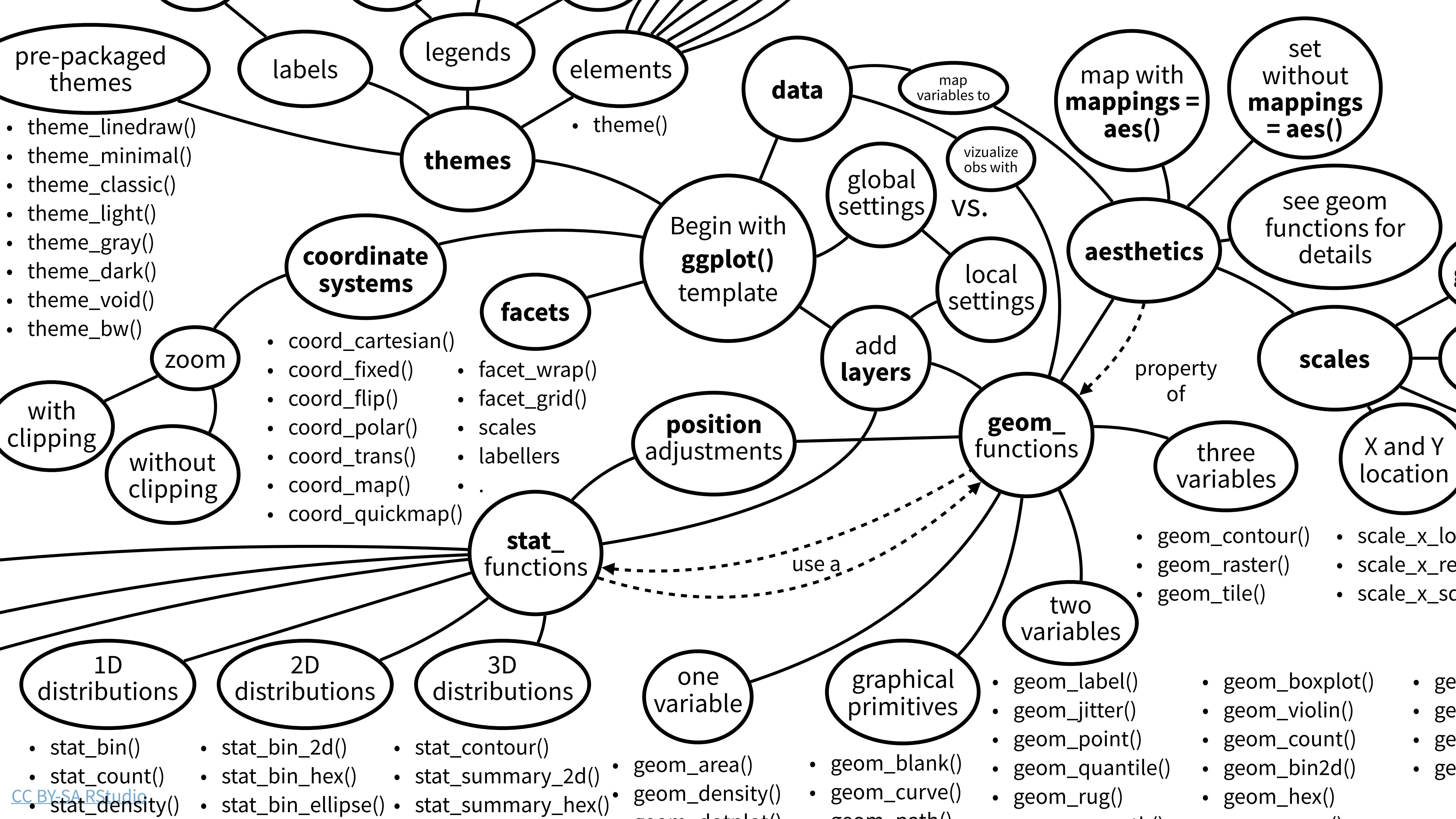
What would a concept map for Tidy Data look like?

Your Turn

Derive a concept map that describes how to make graphs with ggplot2. Possible guides:

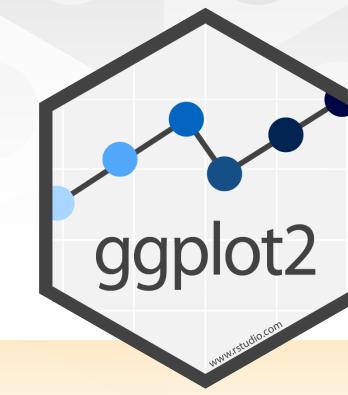
1. 01-Visualize-data slides
2. Data visualization cheatsheet





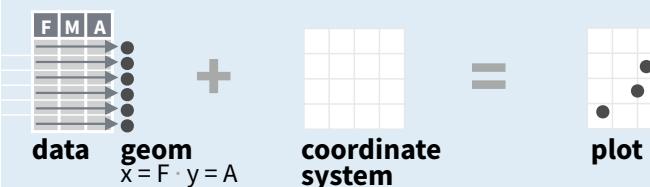
Help > Cheatsheets > Data Visualization with ggplot2

Data Visualization with ggplot2 :: CHEAT SHEET

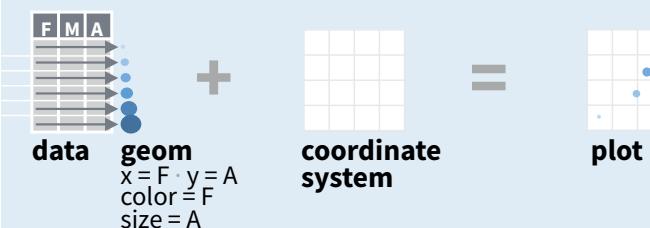


Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and geoms—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot (data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPIINGS>),  
stat = <STAT>, position = <POSITION>) +  
<COORDINATE_FUNCTION> +  
<FACET_FUNCTION> +  
<SCALE_FUNCTION> +  
<THEME_FUNCTION>
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings data geom

qplot(x = cyl, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables.
Each function returns a layer.

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))  
b <- ggplot(seals, aes(x = long, y = lat))  
  
a + geom_blank()  
    # (Useful for expanding limits)  
  
b + geom_curve(aes(yend = lat + 1,  
xend = long + 1, curvature = z)) - x, xend, y, yend,  
alpha, angle, color, curvature, linetype, size  
  
a + geom_path(lineend = "butt", linejoin = "round",  
linemetre = 1)  
x, y, alpha, color, group, linetype, size  
  
a + geom_polygon(aes(group = group))  
x, y, alpha, color, fill, group, linetype, size  
  
b + geom_rect(aes(xmin = long, ymin = lat, xmax =  
long + 1, ymax = lat + 1)) - xmax, xmin, ymax,  
ymin, alpha, color, fill, linetype, size  
  
a + geom_ribbon(aes(ymax = unemploy - 900,  
ymin = unemploy + 900)) - x, ymax, ymin,  
alpha, color, fill, group, linetype, size
```

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

```
b + geom_abline(aes(intercept = 0, slope = 1))  
b + geom_hline(aes(intercept = lat))  
b + geom_vline(aes(xintercept = long))  
  
b + geom_segment(aes(yend = lat + 1, xend = long + 1))  
b + geom_spoke(aes(angle = 1:1155, radius = 1))
```

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)  
  
c + geom_area(stat = "bin")  
x, y, alpha, color, fill, linetype, size  
  
c + geom_density(kernel = "gaussian")  
x, y, alpha, color, fill, group, linetype, size, weight  
  
c + geom_dotplot()  
x, y, alpha, color, fill  
  
c + geom_freqpoly()  
x, y, alpha, color, group, linetype, size  
  
c + geom_histogram(binwidth = 5)  
x, y, alpha, color, fill, linetype, size, weight  
  
c2 + geom_qq(aes(sample = hwy))  
x, y, alpha, color, fill, linetype, size, weight
```

discrete

```
d <- ggplot(mpg, aes(f1))  
  
d + geom_bar()  
x, alpha, color, fill, linetype, size, weight
```

TWO VARIABLES

continuous x , continuous y

```
e <- ggplot(mpg, aes(cty, hwy))  
  
e + geom_label(aes(label = cty), nudge_x = 1,  
nudge_y = 1, check_overlap = TRUE)x, y, label,  
alpha, angle, color, family, fontface, hjust,  
lineheight, size, vjust  
  
e + geom_jitter(height = 2, width = 2)  
x, y, alpha, color, fill, shape, size  
  
e + geom_point(), x, y, alpha, color, fill, shape,  
size, stroke  
  
e + geom_quantile(), x, y, alpha, color, group,  
linetype, size, weight  
  
e + geom_rug(sides = "bl")x, y, alpha, color,  
linetype, size  
  
e + geom_smooth(method = lm), x, y, alpha,  
color, fill, group, linetype, size, weight  
  
e + geom_text(aes(label = cty), nudge_x = 1,  
nudge_y = 1, check_overlap = TRUE)x, y, label,  
alpha, angle, color, family, fontface, hjust,  
lineheight, size, vjust
```

discrete x , continuous y

```
f <- ggplot(mpg, aes(class, hwy))  
  
f + geom_col()  
x, y, alpha, color, fill, group, linetype,  
size  
  
f + geom_boxplot()  
x, y, lower, middle, upper,  
ymin, ymax, alpha, color, fill, group, linetype,  
shape, size, weight  
  
f + geom_dotplot(binaxis = "y", stackdir =  
"center")x, y, alpha, color, fill, group  
  
f + geom_violin(scale = "area")  
x, y, alpha, color, fill, group, linetype, size, weight
```

discrete x , discrete y

```
g <- ggplot(diamonds, aes(cut, color))  
  
g + geom_count()  
x, y, alpha, color, fill, shape,  
size, stroke
```

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))  
l <- ggplot(seals, aes(long, lat))  
  
l + geom_contour(aes(z = z))  
x, y, z, alpha, colour, group, linetype,  
size, weight  
  
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5,  
interpolate = FALSE)  
x, y, alpha, fill  
  
l + geom_tile(aes(fill = z))  
x, y, alpha, color, fill,  
linetype, size, width
```



How to teach ggplot2



How to teach a workshop

1. Teach the **mental model(s)**

Explain them. Practice the skills they support.

2. Put useful details in a **handout**

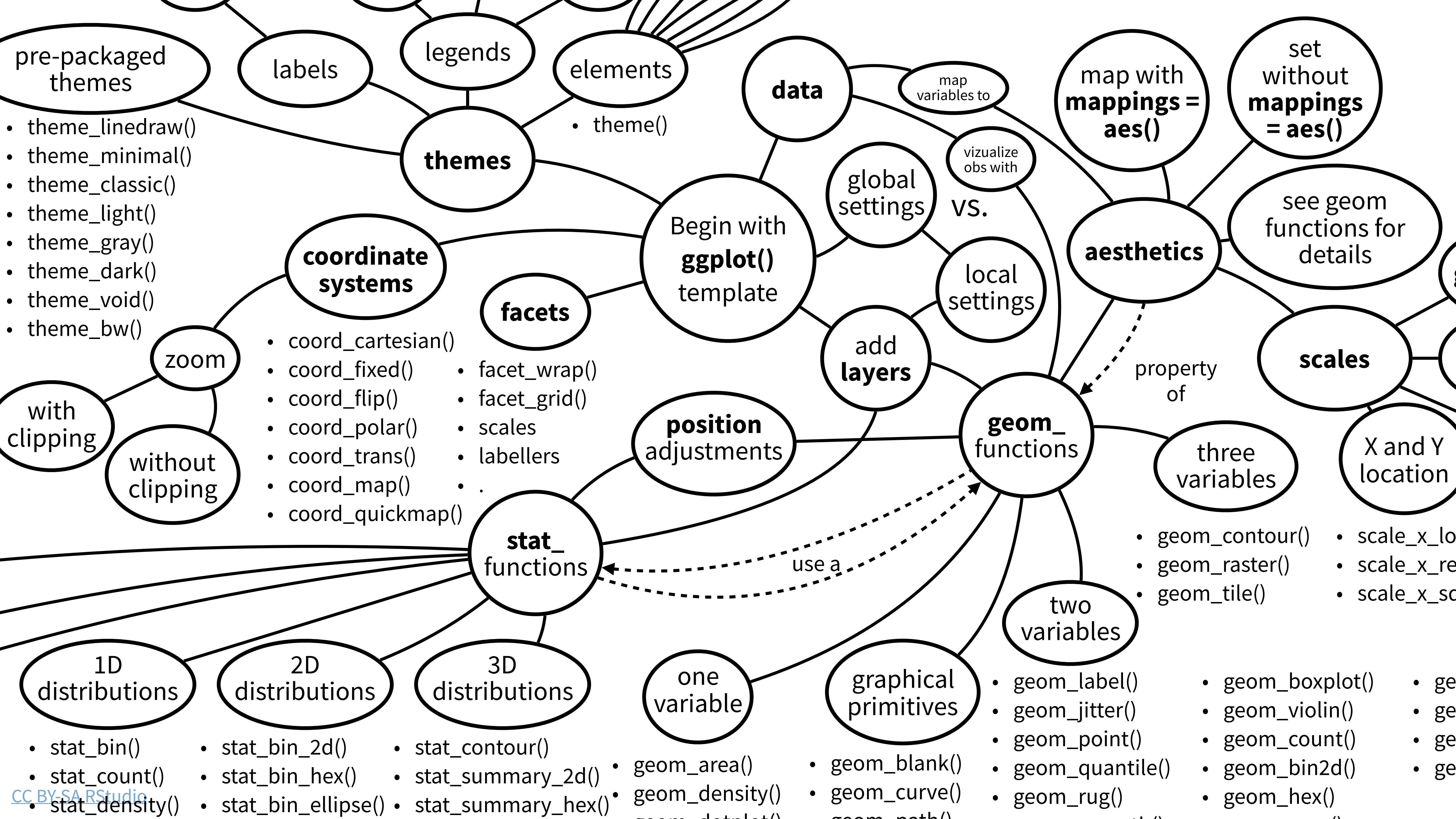
3. Follow up with **post-workshop quizzes**

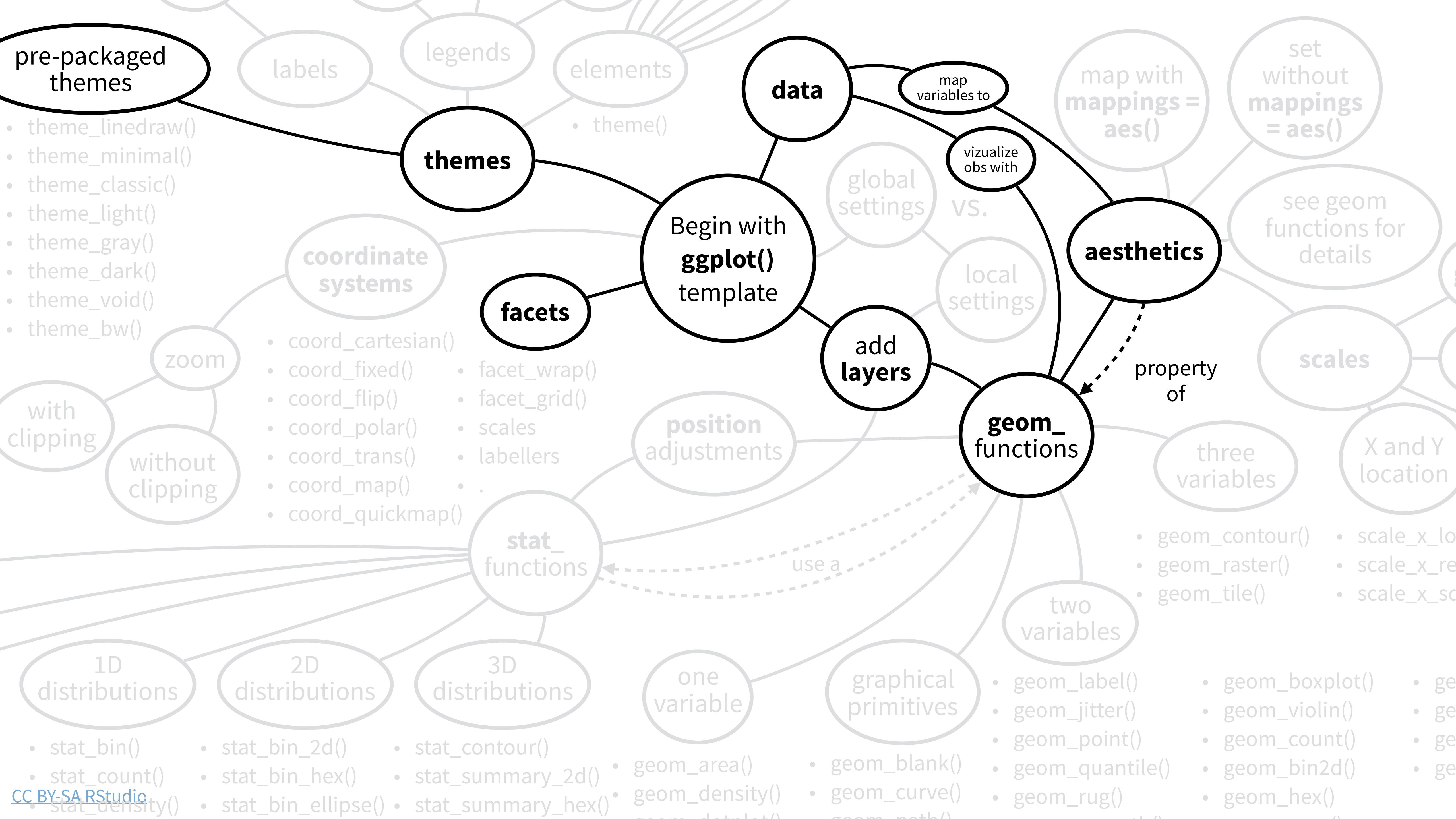
Use cognitive theory as a guide

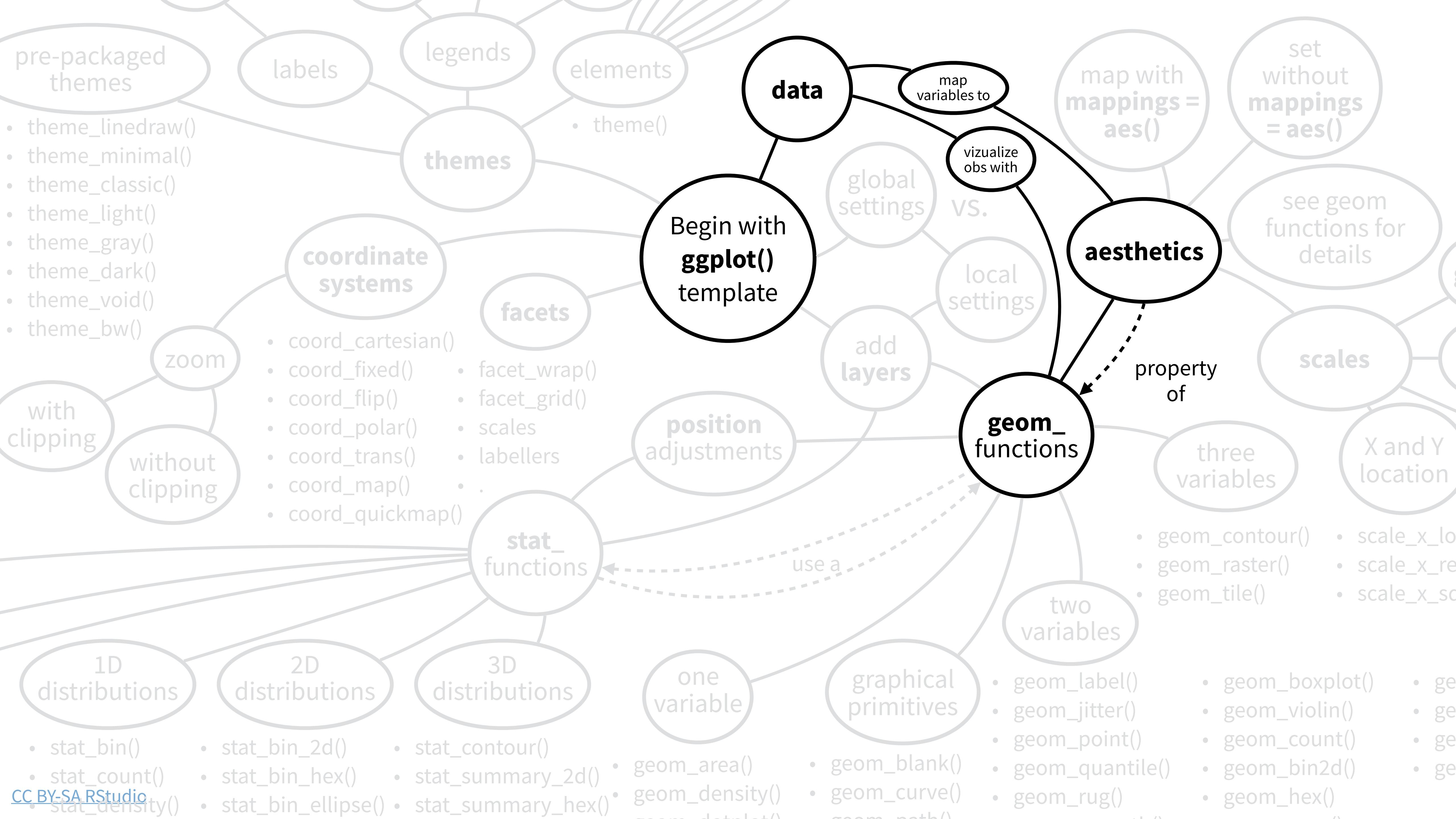
How to teach a workshop

1. Teach the **mental model(s)**

Explain them. Practice the skills they support.







How to teach a workshop

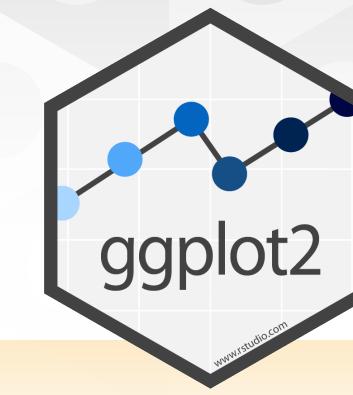
1. Teach the **mental model(s)**

Explain them. Practice the skills they support.

2. Put useful details in a **handout**

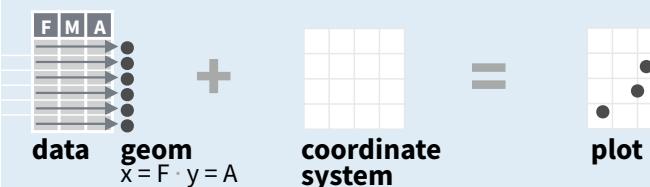
Help > Cheatsheets > Data Visualization with ggplot2

Data Visualization with ggplot2 :: CHEAT SHEET

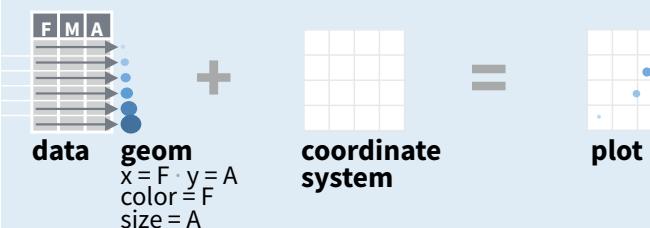


Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data set**, a **coordinate system**, and geoms—visual marks that represent data points.



To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.



Complete the template below to build a graph.

```
ggplot (data = <DATA>) +  
<GEOM_FUNCTION>(mapping = aes(<MAPPIINGS>),  
stat = <STAT>, position = <POSITION>) +  
<COORDINATE_FUNCTION> +  
<FACET_FUNCTION> +  
<SCALE_FUNCTION> +  
<THEME_FUNCTION>
```

ggplot(data = mpg, aes(x = cyl, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings data geom

qplot(x = cyl, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

Geoms

Use a geom function to represent data points, use the geom's aesthetic properties to represent variables.
Each function returns a layer.

GRAPHICAL PRIMITIVES

```
a <- ggplot(economics, aes(date, unemploy))  
b <- ggplot(seals, aes(x = long, y = lat))  
  
a + geom_blank()  
    # (Useful for expanding limits)  
  
b + geom_curve(aes(yend = lat + 1,  
xend = long + 1, curvature = z)) - x, xend, y, yend,  
alpha, angle, color, curvature, linetype, size  
  
a + geom_path(lineend = "butt", linejoin = "round",  
linemetre = 1)  
x, y, alpha, color, group, linetype, size  
  
a + geom_polygon(aes(group = group))  
x, y, alpha, color, fill, group, linetype, size  
  
b + geom_rect(aes(xmin = long, ymin = lat, xmax =  
long + 1, ymax = lat + 1)) - xmax, xmin, ymax,  
ymin, alpha, color, fill, linetype, size  
  
a + geom_ribbon(aes(ymax = unemploy - 900,  
ymin = unemploy + 900)) - x, ymax, ymin,  
alpha, color, fill, group, linetype, size
```

LINE SEGMENTS

common aesthetics: x, y, alpha, color, linetype, size

```
b + geom_abline(aes(intercept = 0, slope = 1))  
b + geom_hline(aes(intercept = lat))  
b + geom_vline(aes(xintercept = long))  
  
b + geom_segment(aes(yend = lat + 1, xend = long + 1))  
b + geom_spoke(aes(angle = 1:1155, radius = 1))
```

ONE VARIABLE continuous

```
c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)  
  
c + geom_area(stat = "bin")  
x, y, alpha, color, fill, linetype, size  
  
c + geom_density(kernel = "gaussian")  
x, y, alpha, color, fill, group, linetype, size, weight  
  
c + geom_dotplot()  
x, y, alpha, color, fill  
  
c + geom_freqpoly()  
x, y, alpha, color, group, linetype, size  
  
c + geom_histogram(binwidth = 5)  
x, y, alpha, color, fill, linetype, size, weight  
  
c2 + geom_qq(aes(sample = hwy))  
x, y, alpha, color, fill, linetype, size, weight
```

discrete

```
d <- ggplot(mpg, aes(f1))  
  
d + geom_bar()  
x, alpha, color, fill, linetype, size, weight
```

TWO VARIABLES

continuous x , continuous y

```
e <- ggplot(mpg, aes(cty, hwy))  
  
e + geom_label(aes(label = cty), nudge_x = 1,  
nudge_y = 1, check_overlap = TRUE)x, y, label,  
alpha, angle, color, family, fontface, hjust,  
lineheight, size, vjust  
  
e + geom_jitter(height = 2, width = 2)  
x, y, alpha, color, fill, shape, size  
  
e + geom_point(), x, y, alpha, color, fill, shape,  
size, stroke  
  
e + geom_quantile(), x, y, alpha, color, group,  
linetype, size, weight  
  
e + geom_rug(sides = "bl")x, y, alpha, color,  
linetype, size  
  
e + geom_smooth(method = lm), x, y, alpha,  
color, fill, group, linetype, size, weight  
  
e + geom_text(aes(label = cty), nudge_x = 1,  
nudge_y = 1, check_overlap = TRUE)x, y, label,  
alpha, angle, color, family, fontface, hjust,  
lineheight, size, vjust
```

discrete x , continuous y

```
f <- ggplot(mpg, aes(class, hwy))  
  
f + geom_col()  
x, y, alpha, color, fill, group, linetype,  
size  
  
f + geom_boxplot()  
x, y, lower, middle, upper,  
ymin, ymax, alpha, color, fill, group, linetype,  
shape, size, weight  
  
f + geom_dotplot(binaxis = "y", stackdir =  
"center")x, y, alpha, color, fill, group  
  
f + geom_violin(scale = "area")  
x, y, alpha, color, fill, group, linetype, size, weight
```

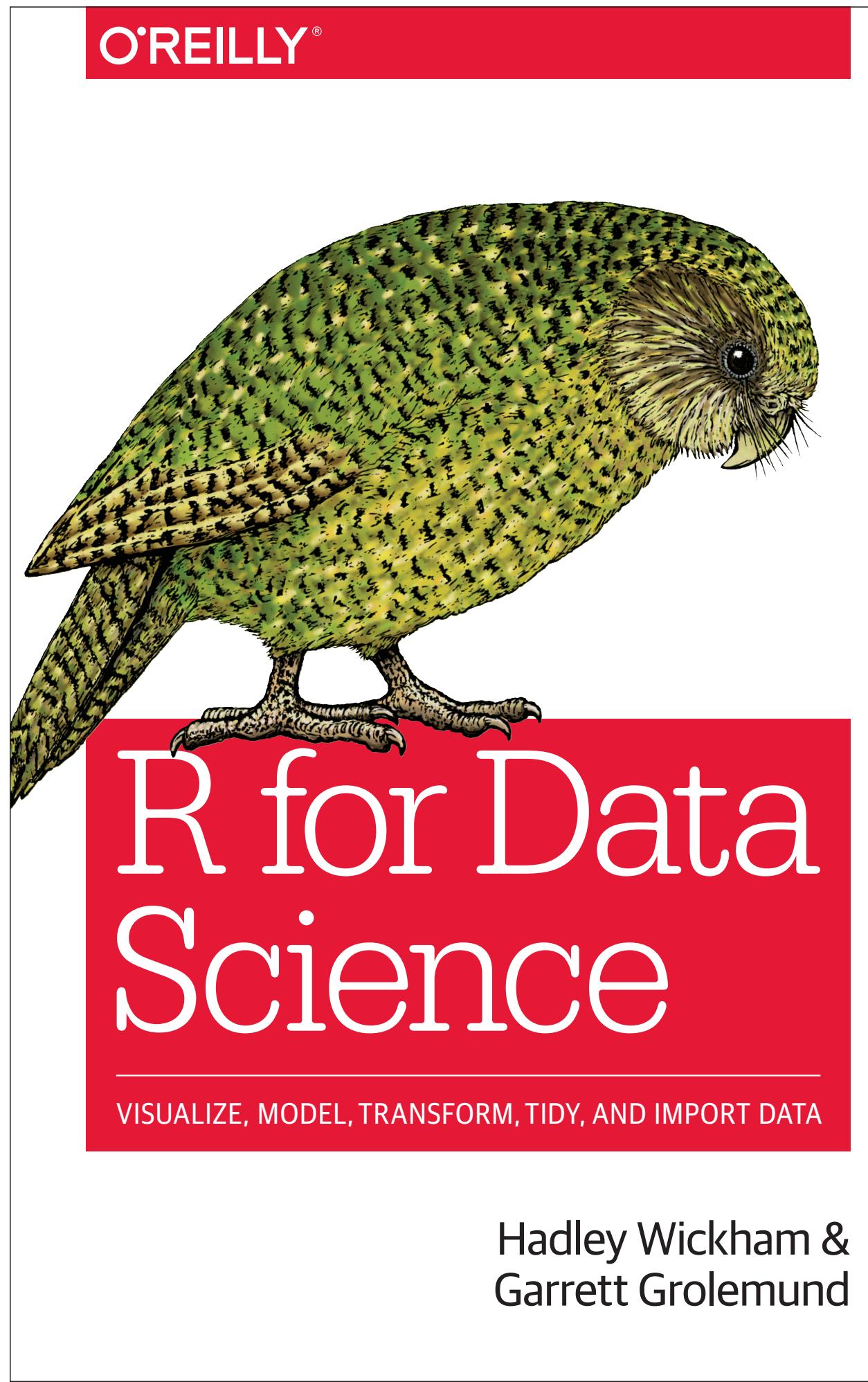
discrete x , discrete y

```
g <- ggplot(diamonds, aes(cut, color))  
  
g + geom_count()  
x, y, alpha, color, fill, shape,  
size, stroke
```

THREE VARIABLES

```
seals$z <- with(seals, sqrt(delta_long^2 + delta_lat^2))  
l <- ggplot(seals, aes(long, lat))  
  
l + geom_contour(aes(z = z))  
x, y, z, alpha, colour, group, linetype,  
size, weight  
  
l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5,  
interpolate = FALSE)  
x, y, alpha, fill  
  
l + geom_tile(aes(fill = z))  
x, y, alpha, color, fill,  
linetype, size, width
```





<http://r4ds.had.co.nz>

R for Data Science

Garrett

>Welcome

1 Introduction

I Explore

2 Introduction

3 Data visualisation

3.1 Introduction

3.2 First steps

3.3 Aesthetic mappings

3.4 Common problems

3.5 Facets

3.6 Geometric objects

3.7 Statistical transformations

3.8 Position adjustments

3.9 Coordinate systems

3.10 The layered grammar of graphics

4 Workflow: basics

5 Data transformation

6 Workflow: scripts

7 Exploratory Data Analysis

8 Workflow: projects

II Wrangle

9 Introduction

10 Tibbles

11 Data import

12 Tidy data

13 Relational data

3 Data visualisation

3.1 Introduction

"The simple graph has brought more information to the data analyst's mind than any other device."
— John Tukey

This chapter will teach you how to visualise your data using ggplot2. R has several systems for making graphs, but ggplot2 is one of the most elegant and most versatile. ggplot2 implements the **grammar of graphics**, a coherent system for describing and building graphs. With ggplot2, you can do more faster by learning one system and applying it in many places.

If you'd like to learn more about the theoretical underpinnings of ggplot2 before you start, I'd recommend reading "The Layered Grammar of Graphics", <http://vita.had.co.nz/papers/layered-grammar.pdf>.

3.1.1 Prerequisites

This chapter focusses on ggplot2, one of the core members of the tidyverse. To access the datasets, help pages, and functions that we will use in this chapter, load the tidyverse by running this code:

```
library(tidyverse)
#> Loading tidyverse: ggplot2
#> Loading tidyverse: tibble
#> Loading tidyverse: tidy
#> Loading tidyverse: readr
#> Loading tidyverse: purrr
#> Loading tidyverse: dplyr
#> Conflicts with tidy packages -----
#>     tidyselect     tidylog
```

How to teach a workshop

1. Teach the **mental model(s)**

Explain them. Practice the skills they support.

2. Put useful details in a **handout**

3. Follow up with **post-workshop quizzes**

RStudio Cloud Garrett

Secure <https://rstudio.cloud/learn/primers/3.5>

Your Workspace 

Visualize Data

Scatterplots

Welcome

Scatterplots

Layers

Coordinate Systems

Start Over

method

You can use the `method` parameter of `geom_smooth()` to fit and display other types of model lines. To do this, pass `method` the name of an R modeling function for `geom_smooth()` to use, such as `lm` (for linear models) or `glm` (for generalized linear models).

In the code below, use `geom_smooth()` to draw the linear model line that fits the data.

Code  Solution  Submit Answer

```
1
```

Continue

How to teach a workshop

1. Teach the **mental model(s)**

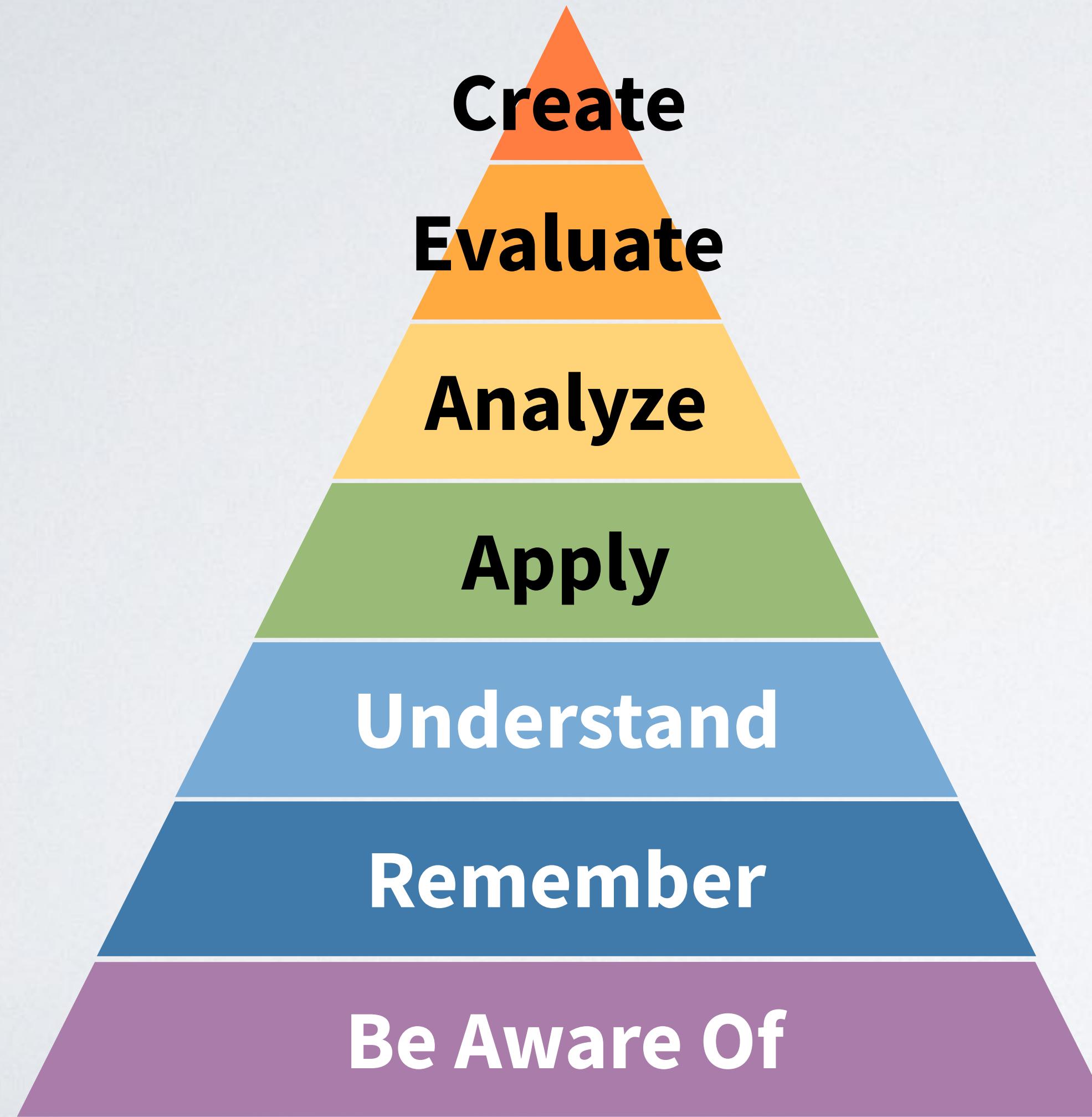
Explain them. Practice the skills they support.

2. Put useful details in a **handout**

3. Follow up with **post-workshop quizzes**

Use cognitive theory as a guide

What do you think?



Find a partner.

One person explain how our prescription helps students attain each level of mastery.

One person explain which levels the prescription neglects or ignores, and why.



Your Turn

Complete the **How to Teach a workshop** handout.



Our Gameplan



How to teach a workshop

1. Teach the **mental model(s)**

Explain them. Practice the skills they support.

2. Put useful details in a **handout**

3. Follow up with **post-workshop quizzes**

Use cognitive theory as a guide

(Receive)

Day 1

Two Ways to Teach

How to Teach a
Workshop

The Cognitive Craft

How to Explain

Make It Clear

Day 2

Their Turn

Motivating Students

How to Practice

Providing R

Make it Stick

(Remember
& Reapply)

Quiz

What is the learning objective for the rest of today?

Quiz

What is the learning objective for tomorrow?

Quiz

What is the prescription for an effective workshop?

How to Teach a Workshop

