

CS 260: Foundations of Data Science

Prof. Sara Mathieson

Spring 2025



HAVERFORD
COLLEGE

- **Lab 6** due Wednesday (tomorrow)
 - In lab today: implementation suggestions and check-ins with everyone
 - Email me if you're completely finished and not coming to lab
- **Lab 7** and **Project Proposal** posted
- Missing lecture/lab without emailing me quickly affects your participation grade!

Outline

- Final project ideas
- Finish logistic regression
- Handout 15
- Introduction to visualization

Outline

- Final project ideas
- Finish logistic regression
- Handout 15
- Introduction to visualization

Timeline and Logistics

- April 2: project proposal due over email
- April 2 - April 29: working on projects
- April 29 and May 1: oral project presentations during class
- May 10 at 5pm: github repos must be finalized for groups with any seniors
- May 16 at 12pm: github repos must be finalized for groups of all non-seniors

Outline for a typical project:

- Find a dataset (see project writeup)
- Run an algorithm we've discussed on the dataset
- Try to do a comparison
 - run the algorithm in multiple ways
 - different data pre-processing
 - try a different algorithm
- Evaluate, interpret, and visualize the results

Project Proposal

- **Title** and **names** of both partners
 - Pair work is required!
- A **dataset** (what is n ? what is p ?)
- An **algorithm** or set of algorithms you will develop and/or apply to this dataset
- A **scientific question** you are trying to answer
 - “Will Naive Bayes or logistic regression perform better on my dataset?”
 - “How will pre-processing a dataset or subsampling features affect the results?”
- A way to **evaluate, interpret, and visualize** the results
- **References**

Project Group Options

- If you would like a random partner, please email me ASAP!
- If you **really** prefer to work individually or in a group of 3, email me ASAP!

Final Project Deliverables

- Main deliverable: presentation
 - In class April 29, May 1 (last week of classes)
 - 10 min per pair
 - Peer feedback
- On git (by May 10 or May 16)
 - Lab Notebook (in README.md)
 - Project Code
 - Slides

Project Lab Notebook

- As you accept your git repo, start creating a “lab notebook” in your **README.md**
- This should say:
 - who was working (which partner)
 - date
 - how long
 - briefly what what accomplished

Sara: 03-07-18 (2hrs)

- now averaging the Markov chain, fixed all the results
- combined ancestral 1000 genomes still running (need to start similar for SGDP)
- started new runs with filtering to only have selected alleles in the “selected pop” and only have ancestral alleles in the “reference panel”

Outline

- Final project ideas
- **Finish logistic regression**
- Handout 15
- Introduction to visualization

Goal: find \vec{w}

$$y \in \{0, 1\}$$

$$\vec{x} \in \mathbb{R}^p$$

Likelihood — want high!

$$L(\vec{w}) = \prod_{i=1}^n \underbrace{h_{\vec{w}}(\vec{x}_i)}_{\text{Prob of 1}}^{y_i} \underbrace{(1 - h_{\vec{w}}(\vec{x}_i))}_{\text{prob of 0}}^{1-y_i}$$

true

$$\vec{y} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

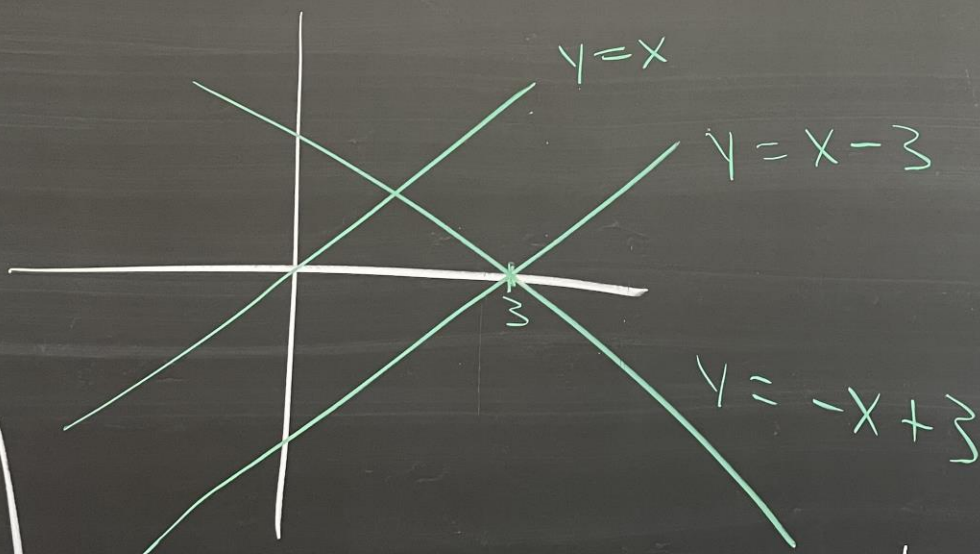
$$n=5$$

$$L(\vec{w}) = h_{\vec{w}}(\vec{x}_1) \cdot h_{\vec{w}}(\vec{x}_2) \cdot (1 - h_{\vec{w}}(\vec{x}_3)) \cdot h_{\vec{w}}(\vec{x}_4) \cdot (1 - h_{\vec{w}}(\vec{x}_5))$$

negative log likelihood \Rightarrow cost

$$J(\vec{w}) = - \sum_{i=1}^n \underbrace{y_i \log h_{\vec{w}}(\vec{x}_i)}_{y_i=1} + \underbrace{(1-y_i) \log (1-h_{\vec{w}}(\vec{x}_i))}_{y_i=0}$$

memorize!



$$\log b^a = a \log b$$

Single example:

$$J(\vec{w}) = \begin{cases} -y \log() \\ -(1-y) \log() \end{cases}$$

$\log(1 - h_{\vec{w}}(\vec{x}_i))$
memorize!

$$h_{\vec{w}}(\vec{x}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}}$$

hint: compute
gradient wrt w

Single example: (\vec{x}, y)

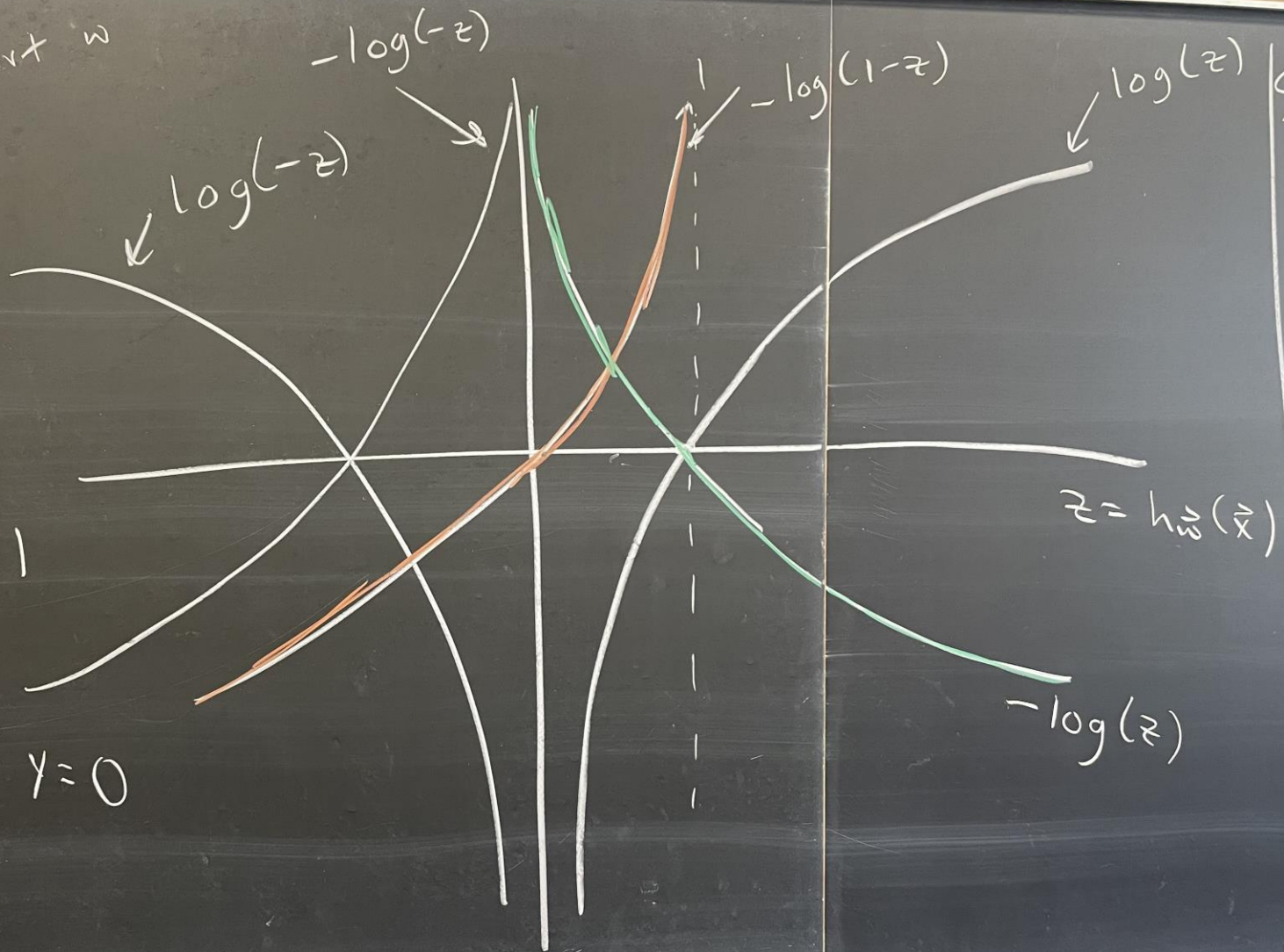
$$J(\vec{w}) = \begin{cases} -y \log(h_{\vec{w}}(\vec{x})) = \underbrace{-\log(h_{\vec{w}}(\vec{x}))}_z & \text{if } y=1 \\ -(1-y) \log(1 - h_{\vec{w}}(\vec{x})) = \underbrace{-\log(1 - h_{\vec{w}}(\vec{x}))}_z & \text{if } y=0 \end{cases}$$

hint: compute
gradient wrt w

$$\underbrace{h_{\vec{w}}(\vec{x})}_{z}$$

if $y=1$

$$\underbrace{1 - h_{\vec{w}}(\vec{x})}_{z} \text{ if } y=0$$

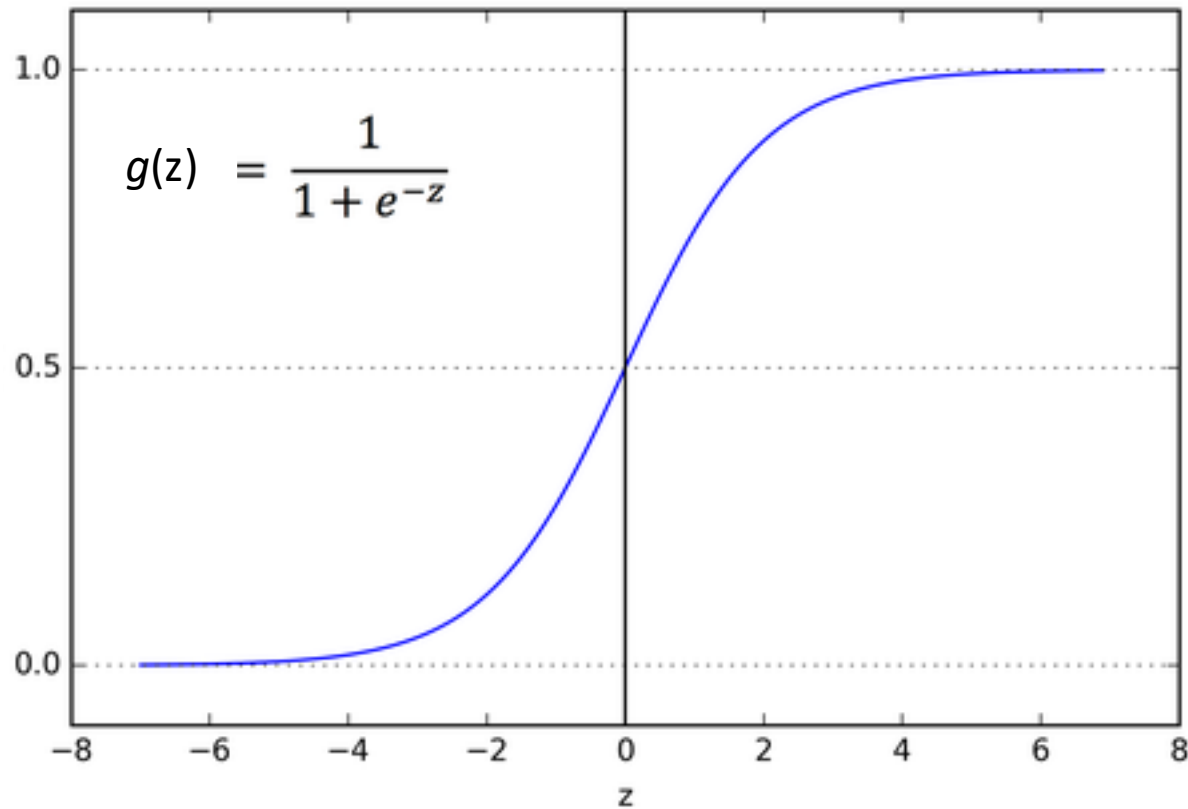


3 important pieces to SGD

- Hypothesis function (prediction)

$$h_w(\mathbf{x}) = p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-w \cdot \mathbf{x}}}$$

Logistic (sigmoid) function



3 important pieces to SGD

- Hypothesis function (prediction)

$$h_{\mathbf{w}}(\mathbf{x}) = p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}}$$

- Cost function (want to minimize)

$$J(\mathbf{w}) = - \sum_{i=1}^n y_i \log h_{\mathbf{w}}(\mathbf{x}_i) + (1 - y_i) \log(1 - h_{\mathbf{w}}(\mathbf{x}_i))$$

3 important pieces to SGD

- Hypothesis function (prediction)

$$h_{\mathbf{w}}(\mathbf{x}) = p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}}$$

- Cost function (want to minimize)

$$J(\mathbf{w}) = - \sum_{i=1}^n y_i \log h_{\mathbf{w}}(\mathbf{x}_i) + (1 - y_i) \log(1 - h_{\mathbf{w}}(\mathbf{x}_i))$$

- Gradient of cost wrt single data point \mathbf{x}_i

$$\nabla J_{\mathbf{x}_i}(\mathbf{w}) = (h_{\mathbf{w}}(\mathbf{x}_i) - y_i) \mathbf{x}_i$$

Stochastic Gradient Descent for Logistic Regression (binary classification)

```
set  $w = 0$  vector
```

```
while cost  $J(w)$  still changing:
```

```
    shuffle data points
```

```
    for  $i = 1 \dots n$ :
```

```
         $w \leftarrow w - \alpha(\text{derivative of } J(w) \text{ wrt } x_i)$ 
```

```
    store  $J(w)$ 
```

SGD

enter loop

for $i = 1 \dots n$:

$$\vec{w} \leftarrow \vec{w} -$$

$$\alpha \nabla J_{\vec{x}_i}(\vec{w})$$

cost for one example

many steps! (exercise)

$$\vec{w} \leftarrow \vec{w} -$$

$$\alpha (h_{\vec{w}}(\vec{x}_i) - y_i) \vec{x}_i$$

Same as
linear
regression!

if equal

\vec{w} doesn't
change!

but $h_{\vec{w}}(\vec{x}_i)$
is different

Cost function as Cross Entropy

$$J(\vec{w}) = -y \log h(x) - (1-y) \log (1-h_w(x))$$

binary cross-entropy

one datapoint

Cross entropy

$$H(p, q) = - \sum_x p(x) \log(q(x))$$

↑
Probability
distribution

ex

true

$$y = 0, 1 - y = 1$$

pred

$$h = 0.4, 1 - h = 0.6$$

one-hot

class0 class1

$$y = [1, 0]$$

$$h = [0.6, 0.4]$$



Outline

- Final project ideas
- Finish logistic regression
- **Handout 15**
- Introduction to visualization

Handout 15

1. The output of logistic regression is a model that creates:
 - (a) a linear decision boundary
 - (b) a logistic decision boundary
 - (c) no decision boundary

Handout 15

1. The output of logistic regression is a model that creates:

- (a) a linear decision boundary
- (b) a logistic decision boundary
- (c) no decision boundary

2. We use logistic regression for:

- (a) classification
- (b) regression
- (c) both

Handout 15

1. The output of logistic regression is a model that creates:

- (a) a linear decision boundary
- (b) a logistic decision boundary
- (c) no decision boundary

2. We use logistic regression for:

- (a) classification
- (b) regression
- (c) both

3. Our hypothesis in logistic regression is:

$$h_{\mathbf{w}}(\mathbf{x}) = p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}}$$

If \mathbf{w} is the zero vector (as it would be when starting SGD), what is the probability $y = 1$?

Handout 15

1. The output of logistic regression is a model that creates:

- (a) a linear decision boundary
- (b) a logistic decision boundary
- (c) no decision boundary

2. We use logistic regression for:

- (a) classification
- (b) regression
- (c) both

3. Our hypothesis in logistic regression is:

$$h_{\mathbf{w}}(\mathbf{x}) = p(y = 1|\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{x}}}$$

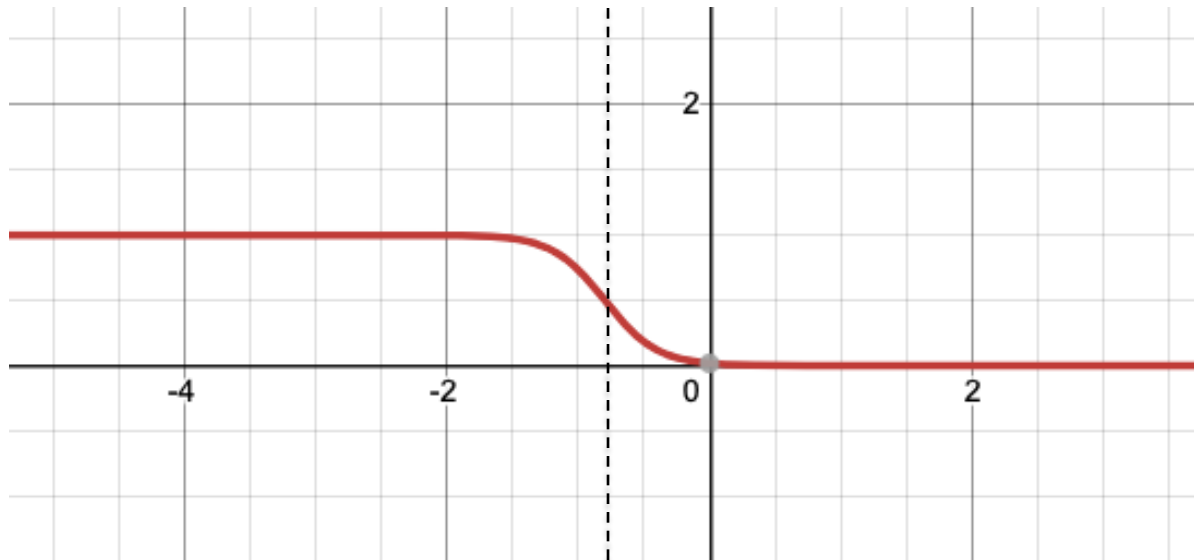
If \mathbf{w} is the zero vector (as it would be when starting SGD), what is the probability $y = 1$?

$\frac{1}{2}$

Handout 15

Say I train a binary logistic regression model (i.e. outcomes $\in \{0, 1\}$) and end up with $\hat{\mathbf{w}} = [\hat{w}_0, \hat{w}_1]^T = [-4, -5]^T$. What is the decision boundary? Sketch a graph of this logistic model and label the decision boundary. How would you classify a new point $x_{\text{test}} = -2$?

$$\frac{1}{1 + e^{-(-4 - 5x)}}$$



Decision boundary

$$h_{\vec{w}}(\vec{x}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{x}}} \geq \frac{1}{2}$$

prob of class 1 \Rightarrow predict $\hat{y} = 1$

$$2 \geq 1 + e^{-\vec{w} \cdot \vec{x}}$$

$$\log(1) \geq \log(e^{-\vec{w} \cdot \vec{x}})$$

$$0 \geq -\vec{w} \cdot \vec{x}$$

$$\boxed{\vec{w} \cdot \vec{x} \geq 0} \Rightarrow \hat{y} = 1$$

now solve for x

$$(4) \quad \vec{w} = \begin{bmatrix} -4 \\ -5 \end{bmatrix} \quad \vec{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

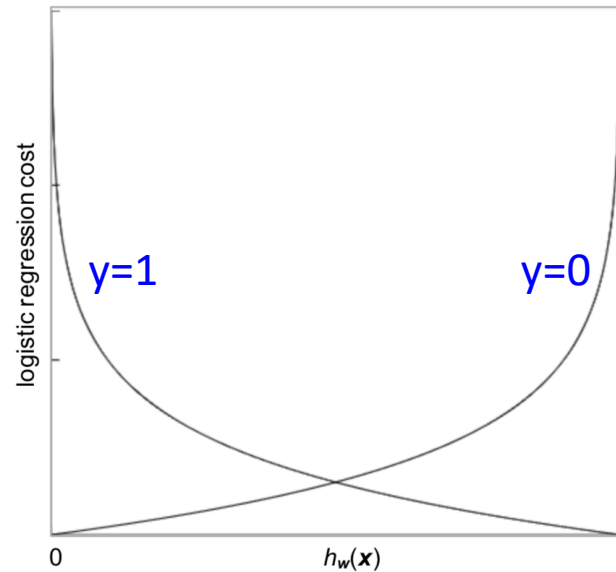
$$\vec{w} \cdot \vec{x} = -4 - 5x \geq 0$$

$$-5x \geq 4$$

$$\boxed{x \leq -\frac{4}{5}} \Rightarrow \hat{y} = 1$$

Handout 15

5. The graph below shows the cost for logistic regression as a function of the hypothesis $h_w(\mathbf{x})$, for one example \mathbf{x} . Which curve corresponds to the true label $y = 0$ and which corresponds to $y = 1$?



Outline

- Final project ideas
- Finish logistic regression
- Handout 15
- Introduction to visualization

Ugly, bad, wrong visualizations

- **ugly**—A figure that has aesthetic problems but otherwise is clear and informative.
- **bad**—A figure that has problems related to perception; it may be unclear, confusing, overly complicated, or deceiving.
- **wrong**—A figure that has problems related to mathematics; it is objectively incorrect.

Ugly, bad, wrong visualizations

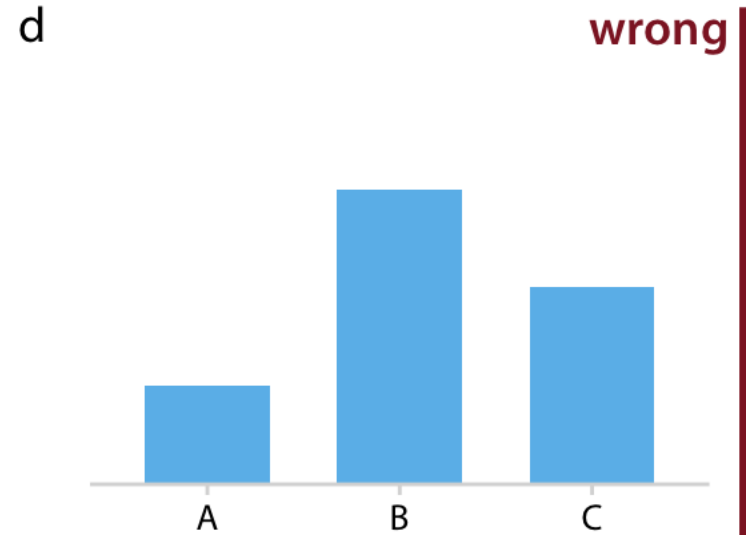
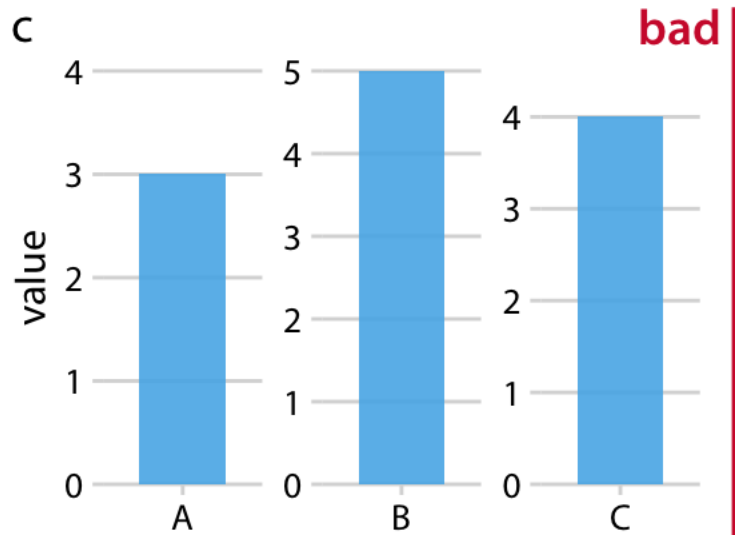
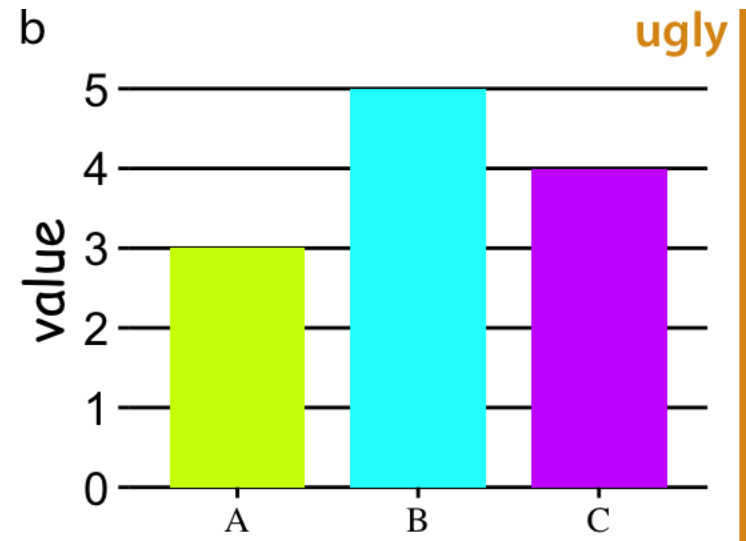
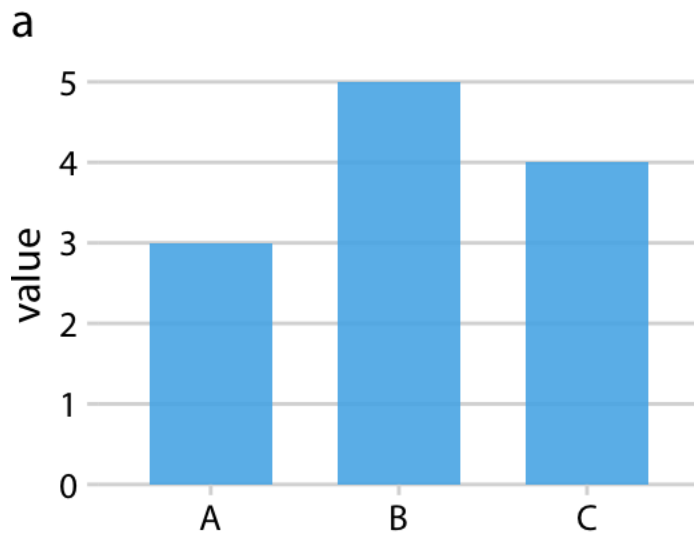


Fig 1.1 from "Fundamentals of Data Visualization" by Claus Wilke

Data Types: continuous vs. discrete

Table 2.1: Types of variables encountered in typical data visualization scenarios.

Type of variable	Examples	Appropriate scale	Description
quantitative/numerical continuous	1.3, 5.7, 83, 1.5×10^{-2}	continuous	Arbitrary numerical values. These can be integers, rational numbers, or real numbers.
quantitative/numerical discrete	1, 2, 3, 4	discrete	Numbers in discrete units. These are most commonly but not necessarily integers. For example, the numbers 0.5, 1.0, 1.5 could also be treated as discrete if intermediate values cannot exist in the given dataset.
qualitative/categorical unordered	dog, cat, fish	discrete	Categories without order. These are discrete and unique categories that have no inherent order. These variables are also called <i>factors</i> .
qualitative/categorical ordered	good, fair, poor	discrete	Categories with order. These are discrete and unique categories with an order. For example, “fair” always lies between “good” and “poor”. These variables are also called <i>ordered factors</i> .
date or time	Jan. 5 2018, 8:03am	continuous or discrete	Specific days and/or times. Also generic dates, such as July 4 or Dec. 25 (without year).
text	The quick brown fox jumps over the lazy dog.	none, or discrete	Free-form text. Can be treated as categorical if needed.

Aesthetics in Data Visualization

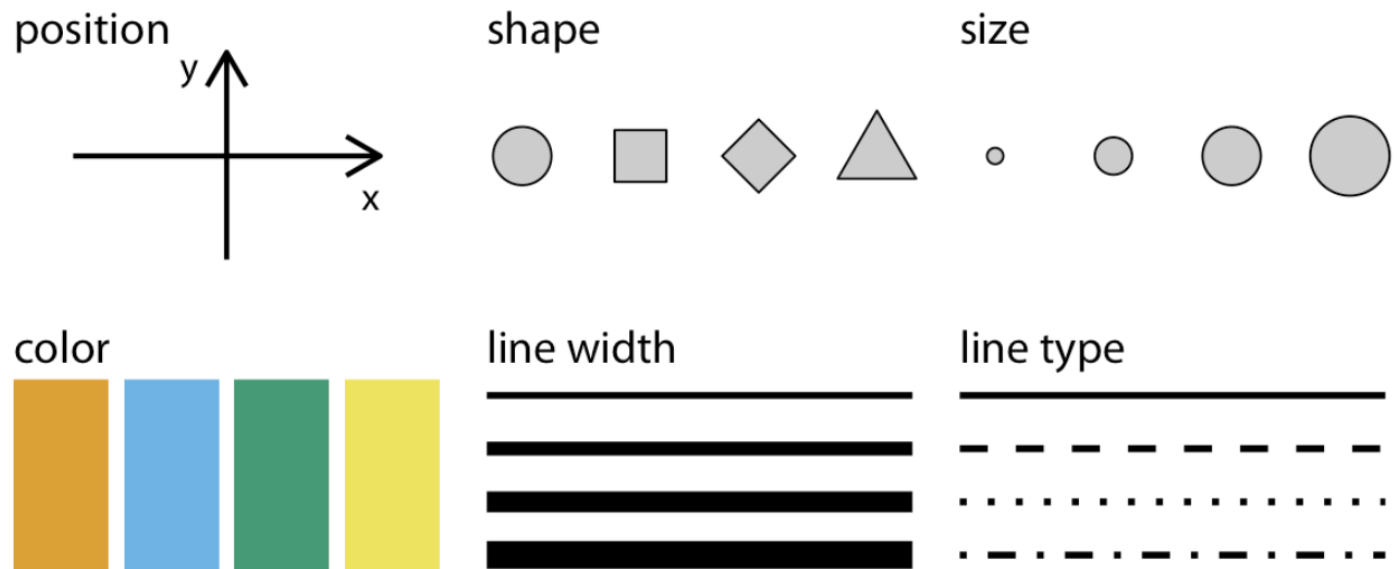
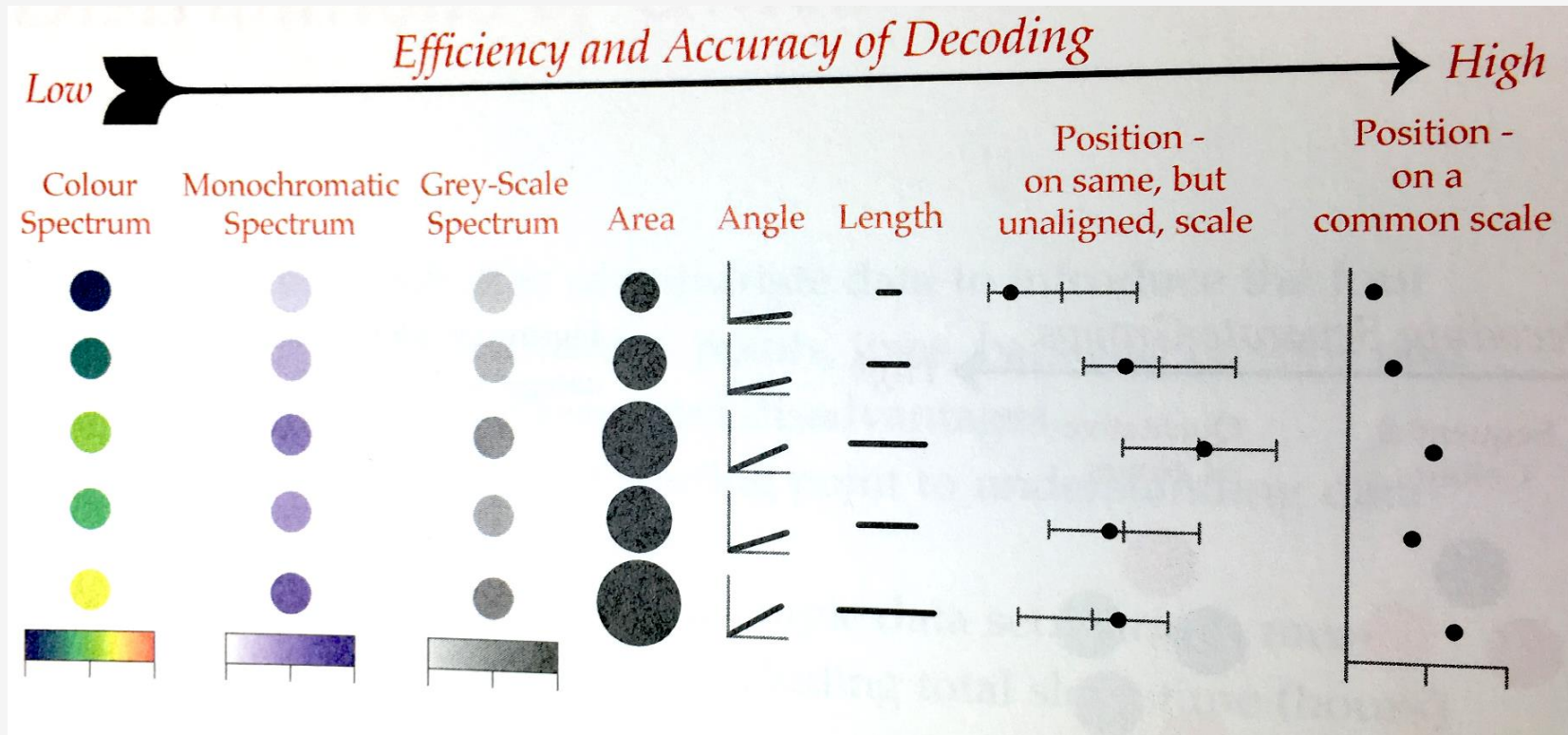


Figure 2.1: Commonly used aesthetics in data visualization: position, shape, size, color, line width, line type. Some of these aesthetics can represent both continuous and discrete data (position, size, line width, color) while others can usually only represent discrete data (shape, line type).

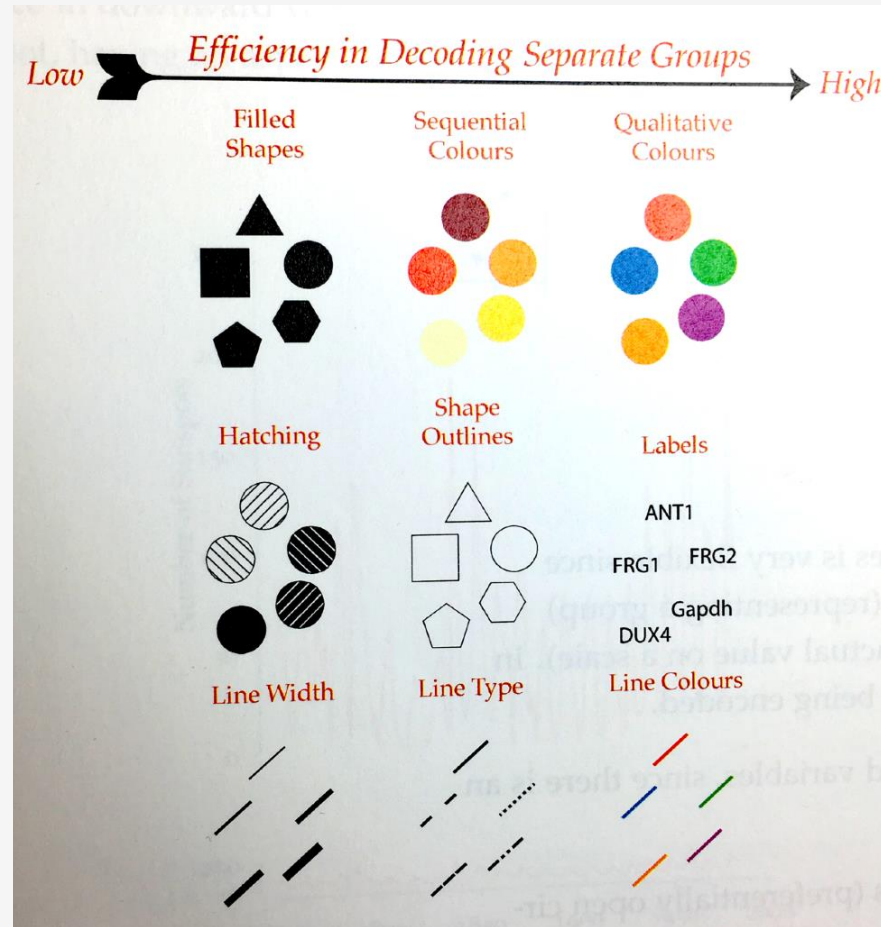
Data Types

Continuous



Data Types

Discrete



Two different visualizations of the same data

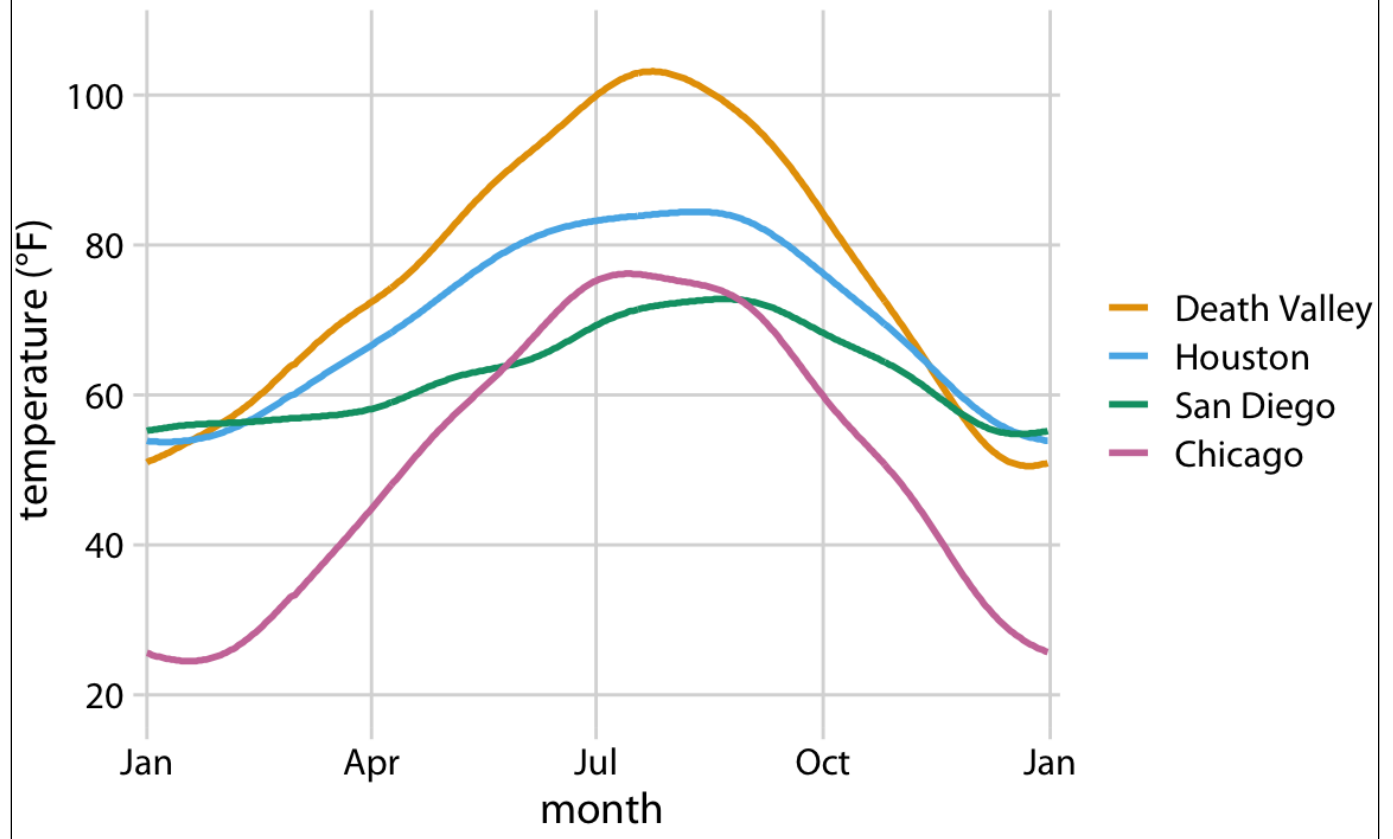
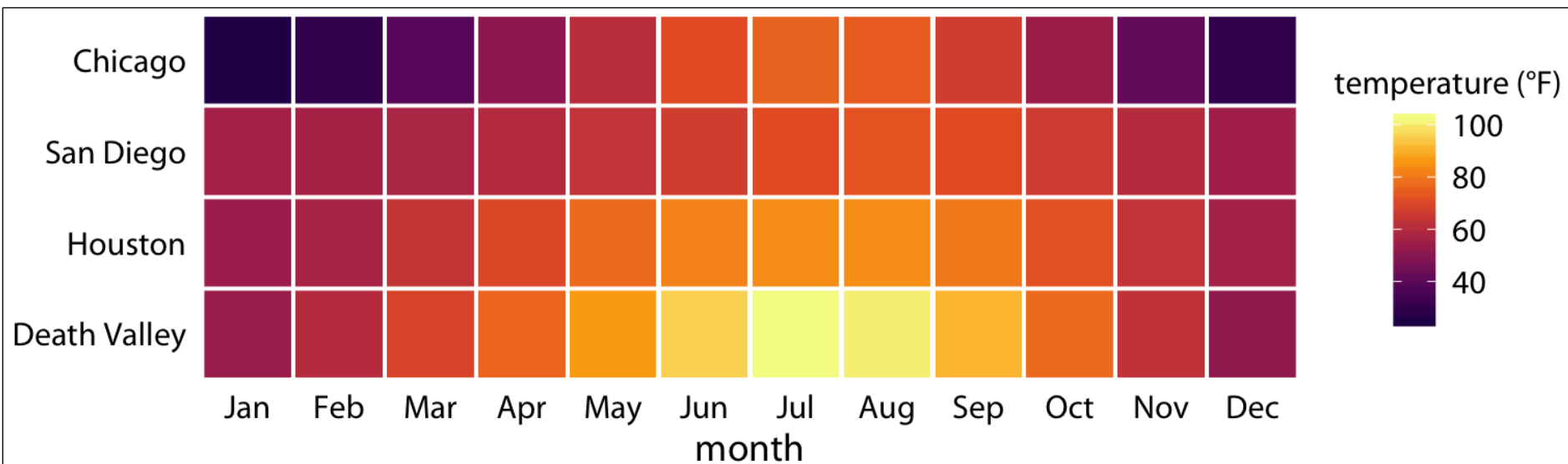


Fig 2.3/2.4 from
“Fundamentals of Data
Visualization” by Claus Wilke



All the same data – what do we want to convey?

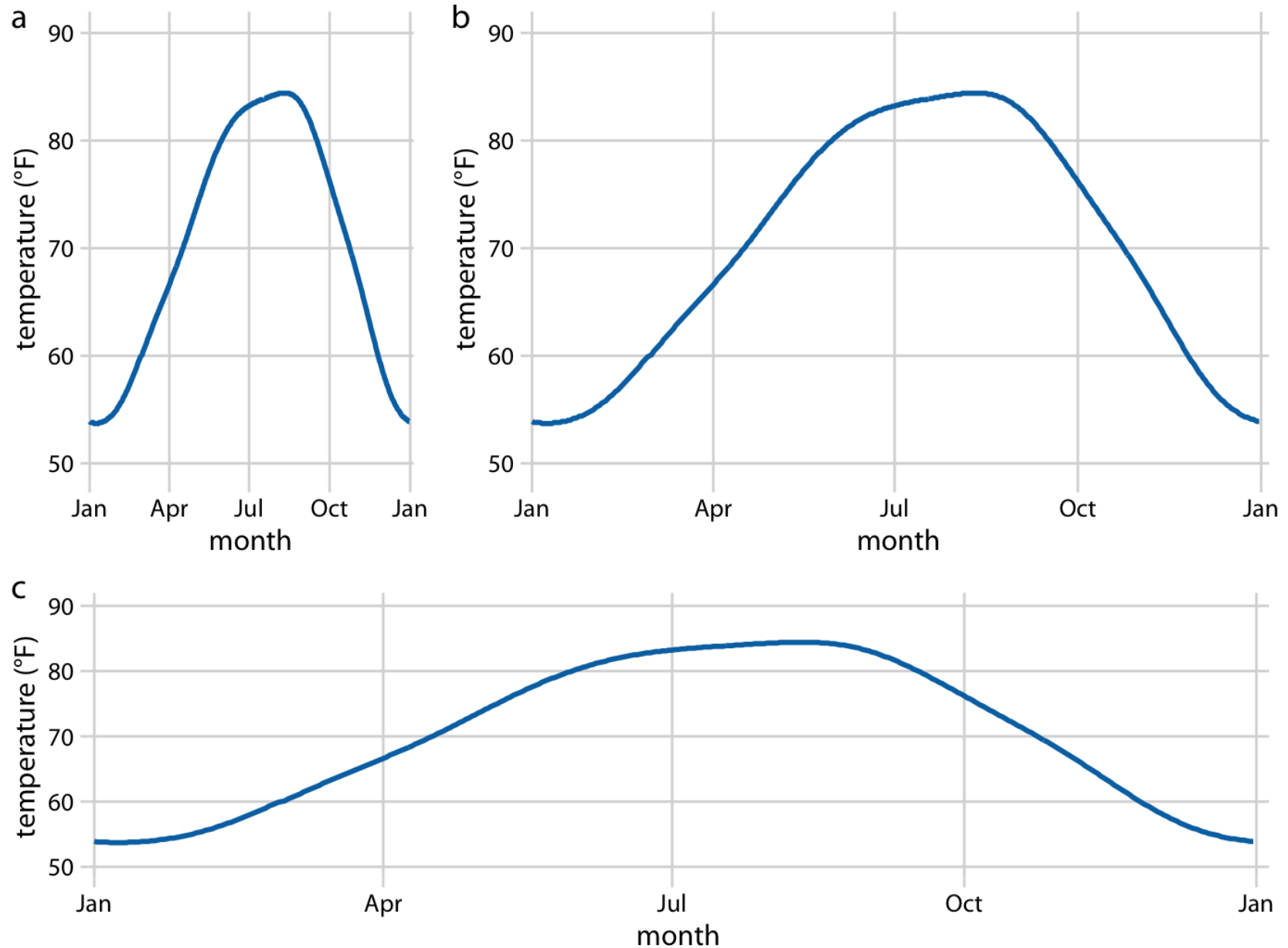
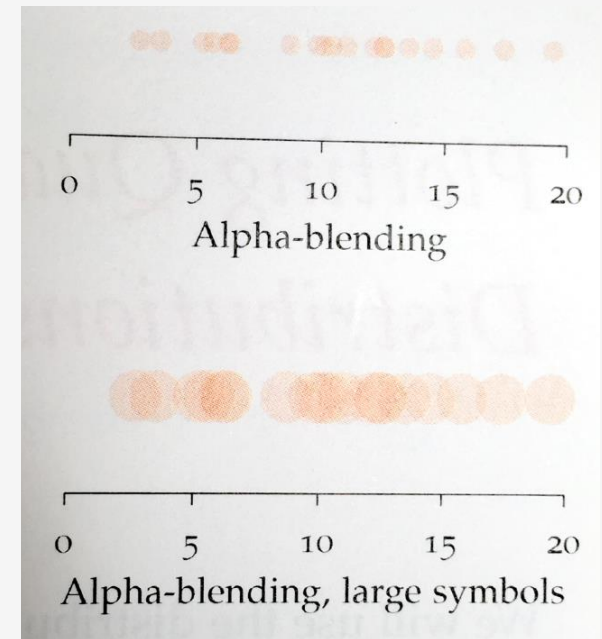
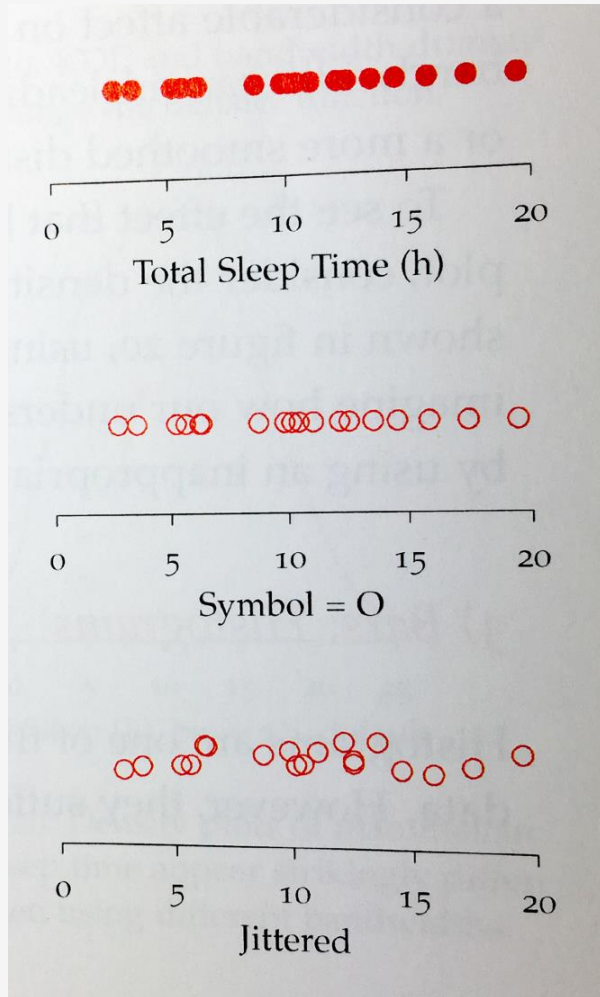


Fig 3.2 from “Fundamentals of Data Visualization” by Claus Wilke

Overplotting



Color

Why use color?

Which colors?

Color: qualitative

Okabe Ito



ColorBrewer Dark2



ggplot2 hue



Qualitative example

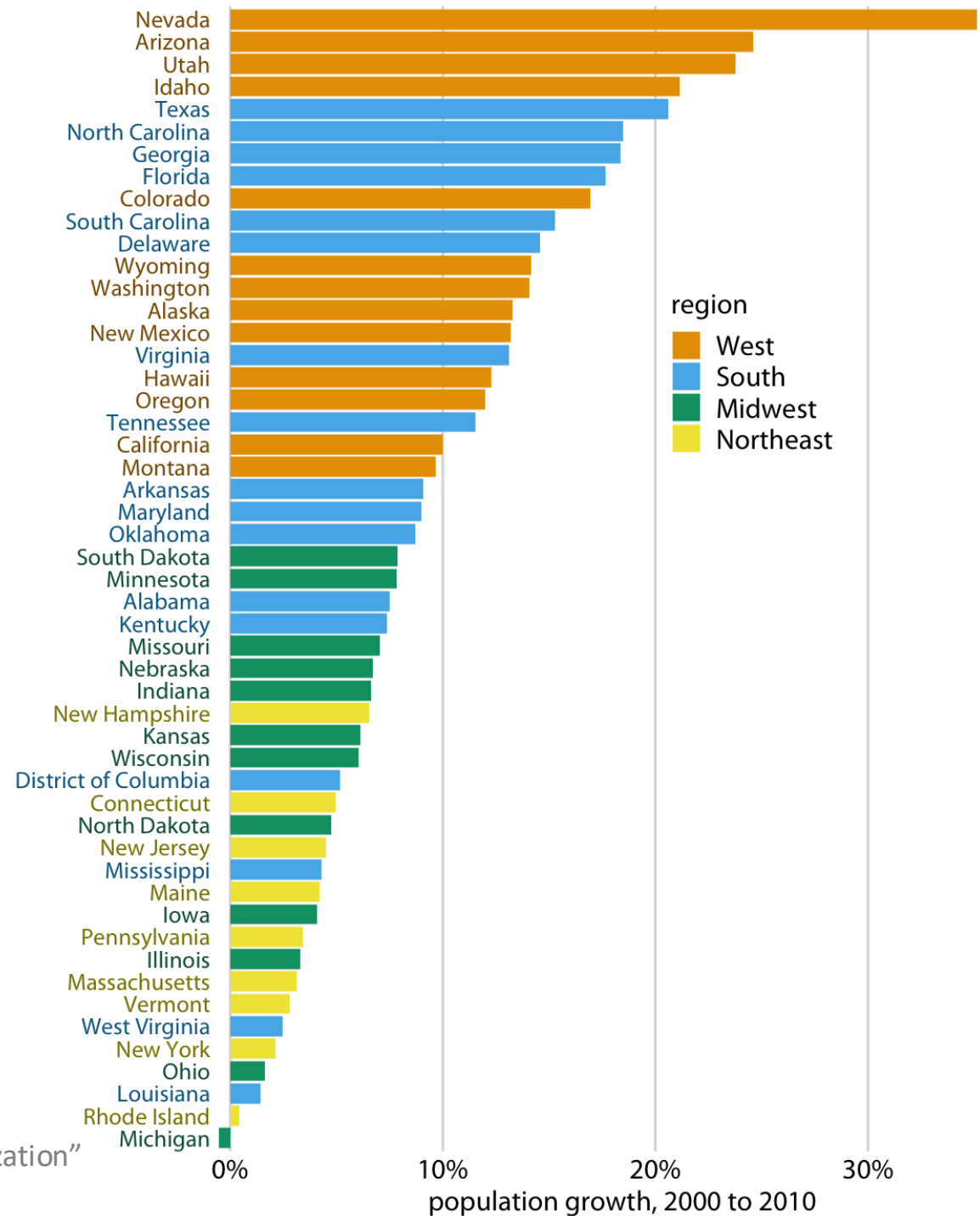


Fig 4.2 from "Fundamentals of Data Visualization"
by Claus Wilke

Color: sequential

ColorBrewer Blues



Heat



Viridis



Sequential example

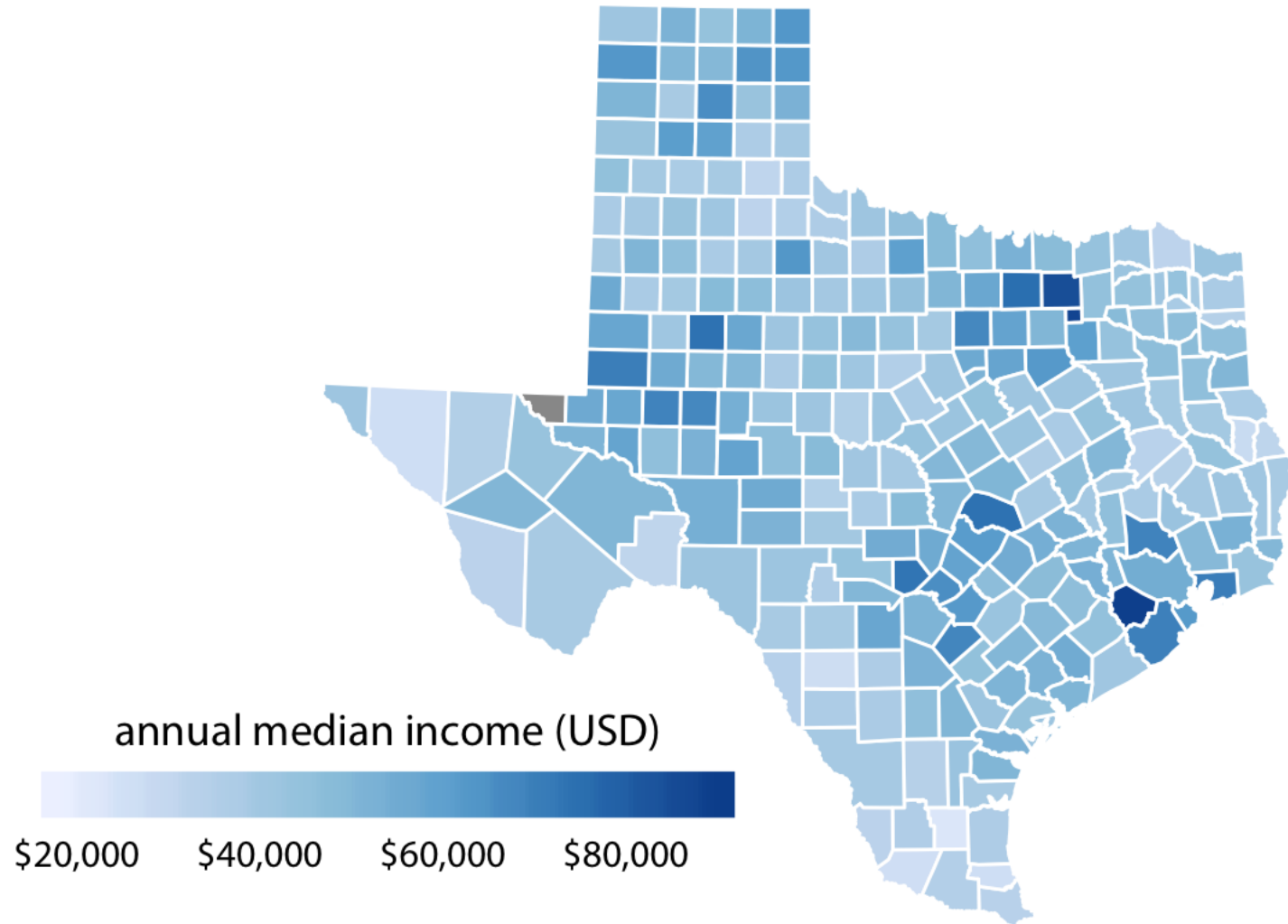


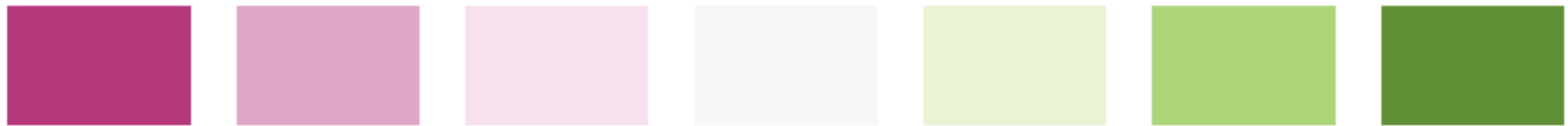
Fig 4.4 from "Fundamentals of Data Visualization" by Claus Wilke

Color: diverging

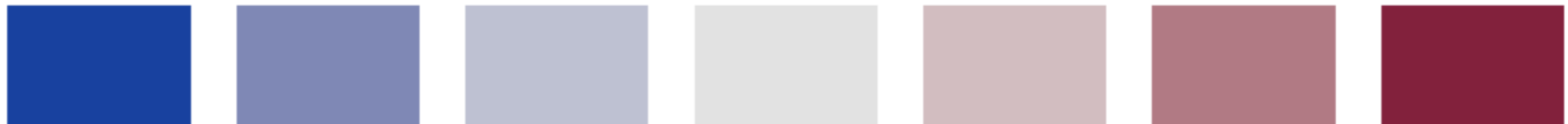
CARTO Earth



ColorBrewer PiYG



Blue-Red



Diverging example

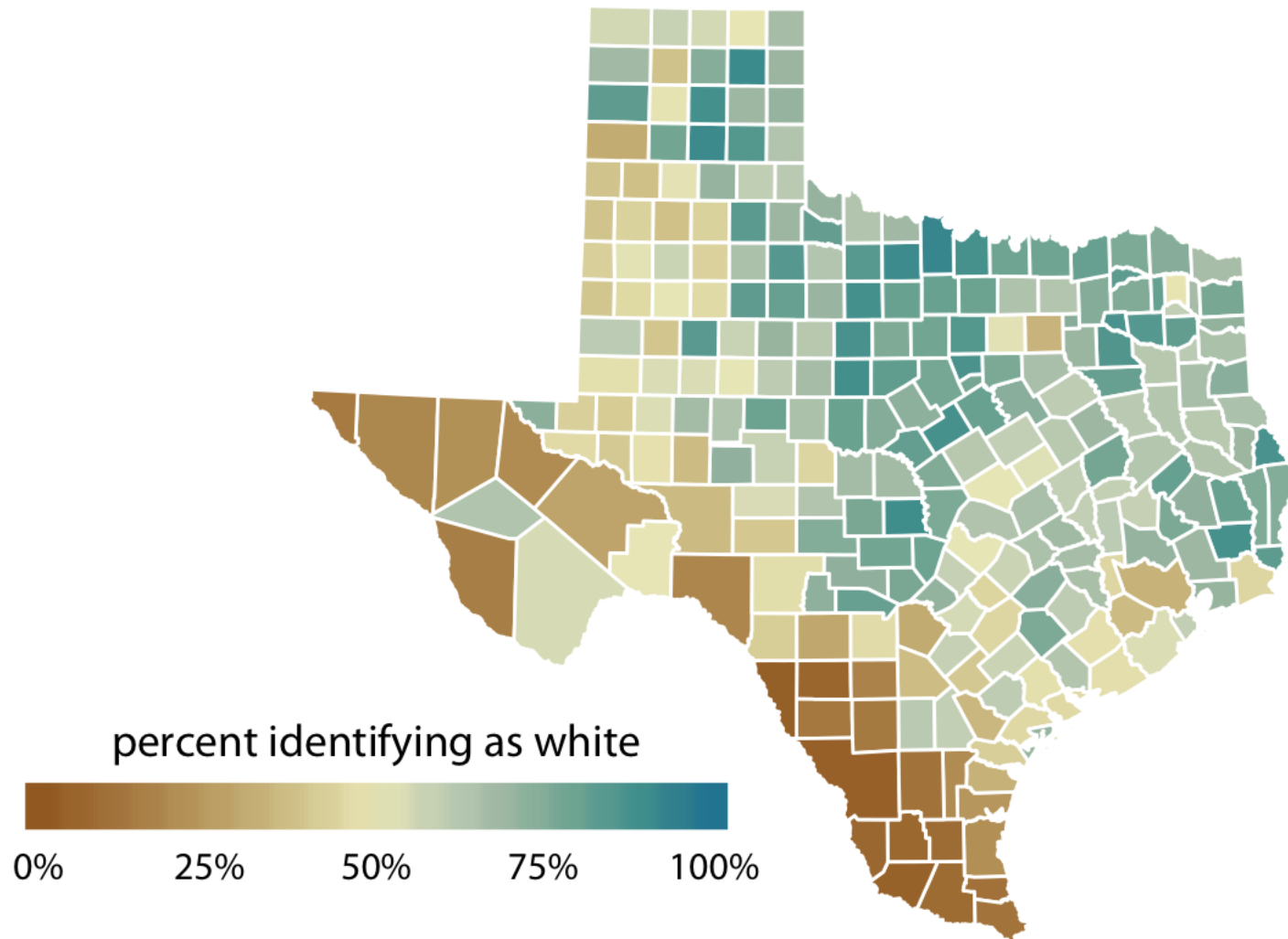
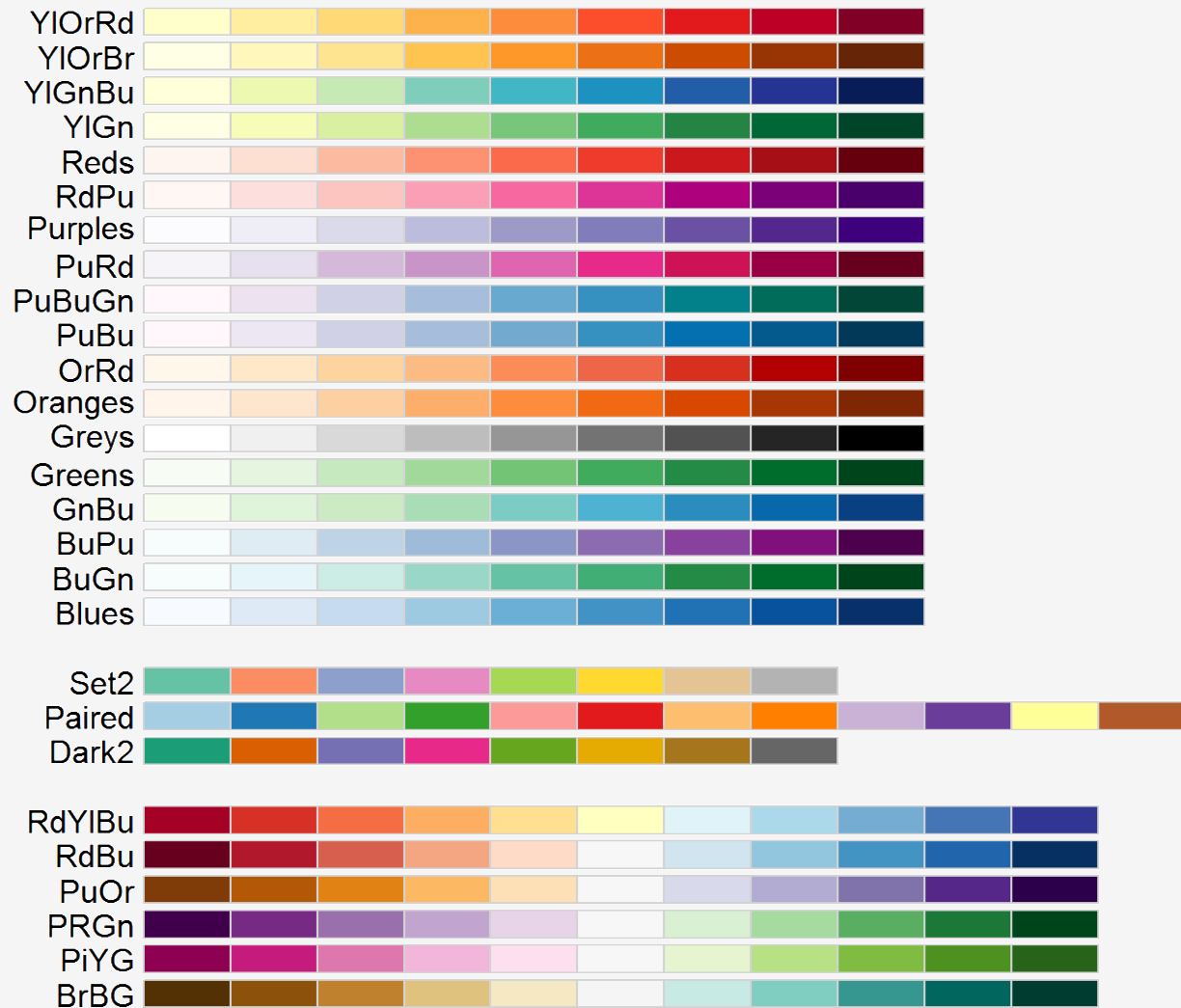


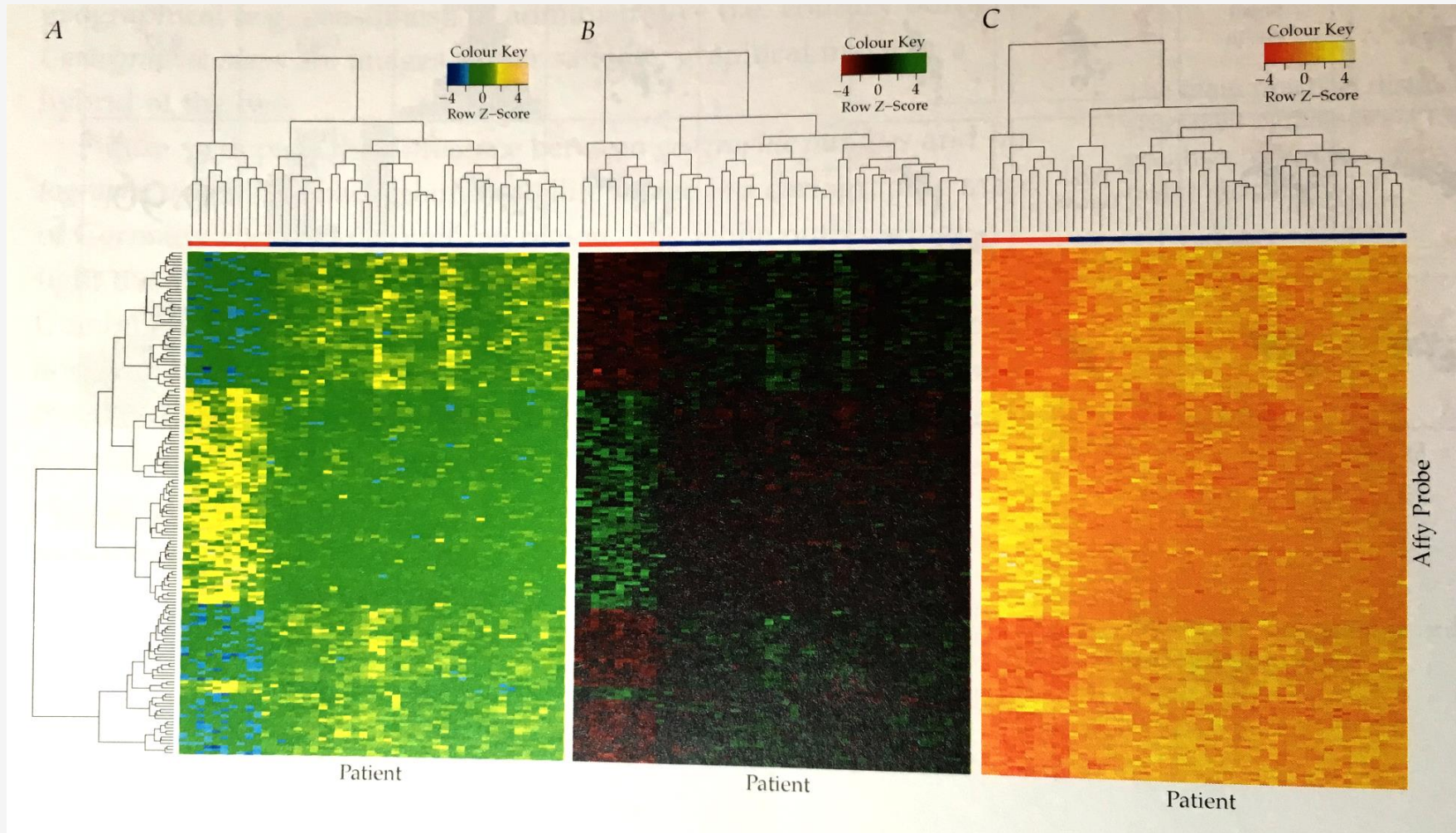
Fig 4.6 from "Fundamentals of Data Visualization" by Claus Wilke

Color

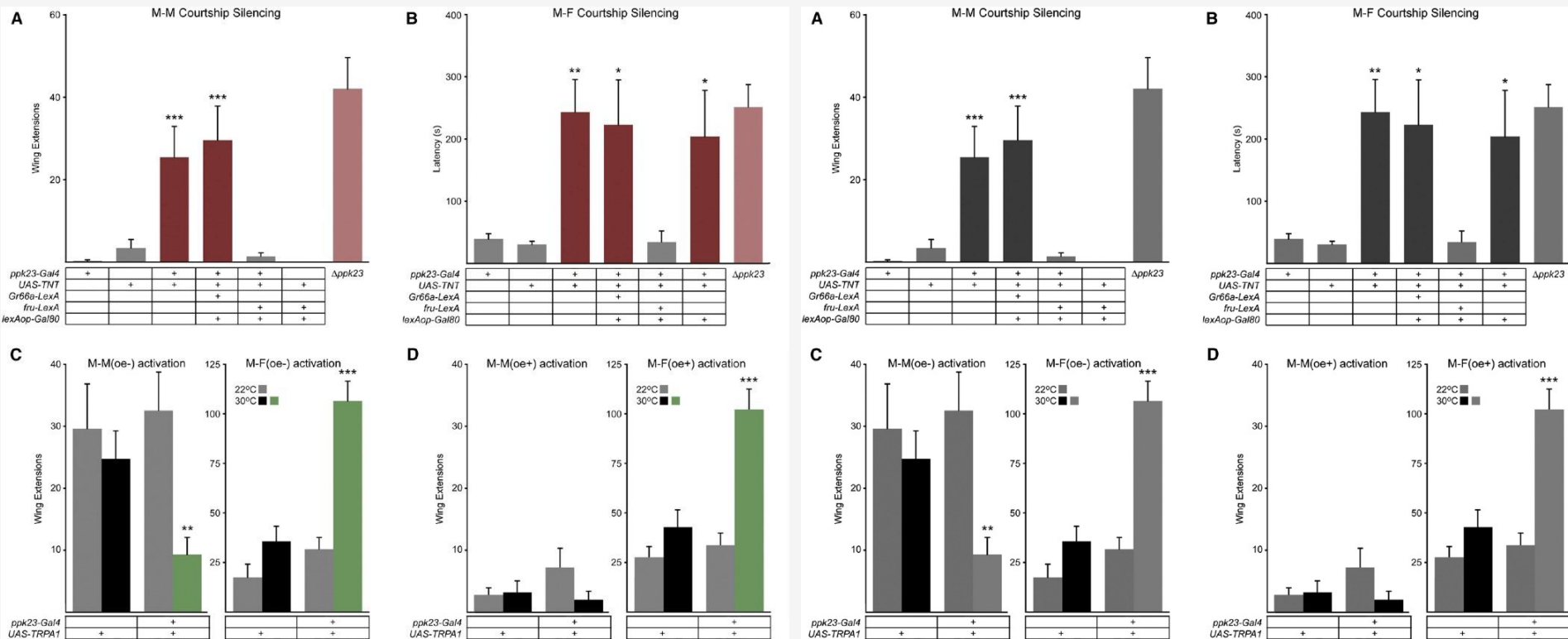
Color-blind friendly



Color

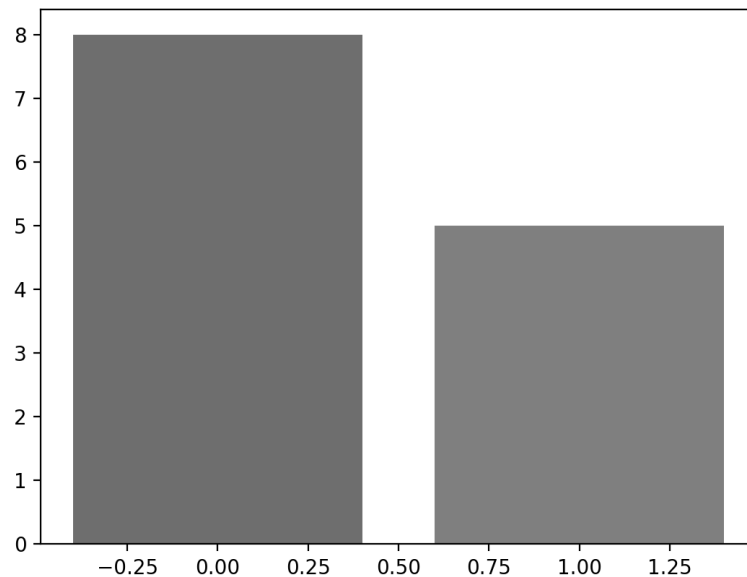
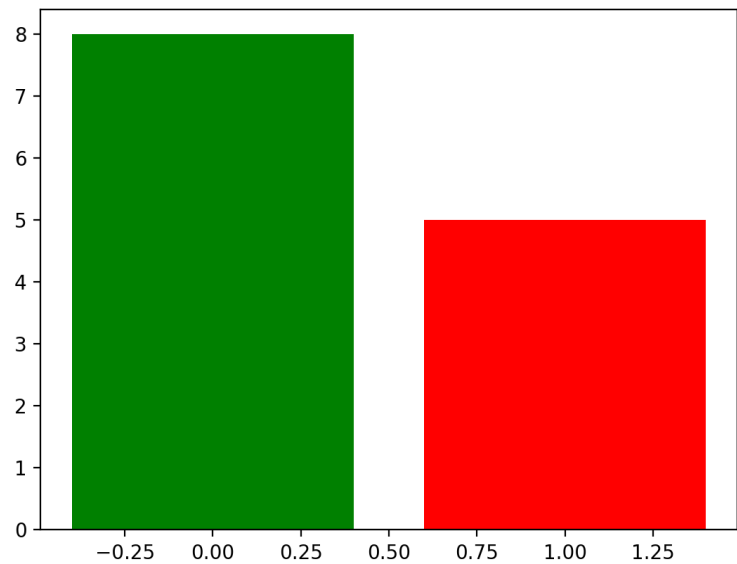


Color

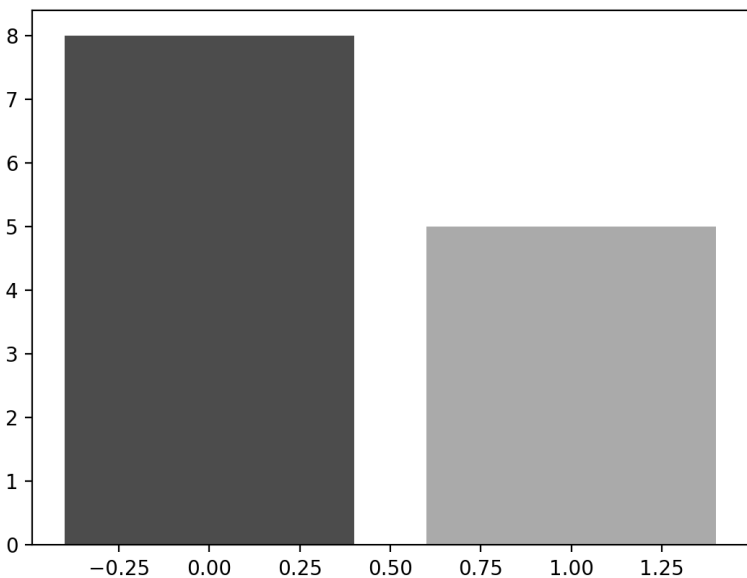
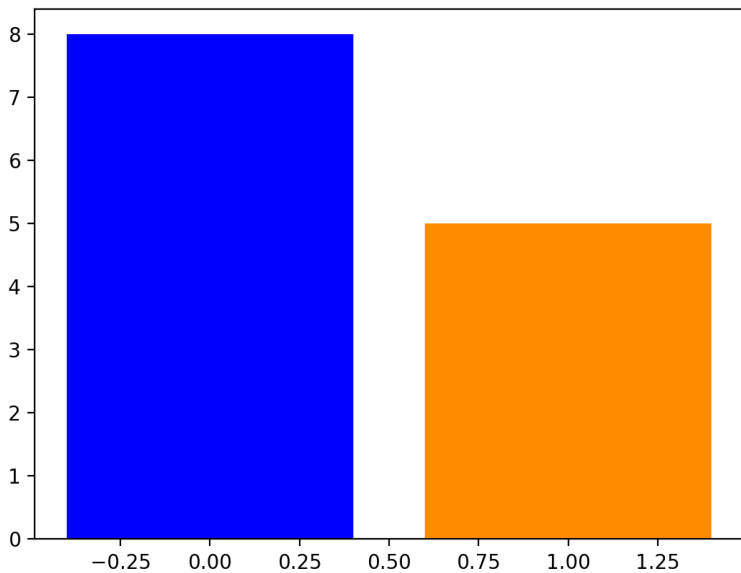


A) Red bars denote responses different than ppk23-Gal4 control.
 B) Red bars denote responses different than ppk23-Gal4 control.
 C) ???
 D) ???

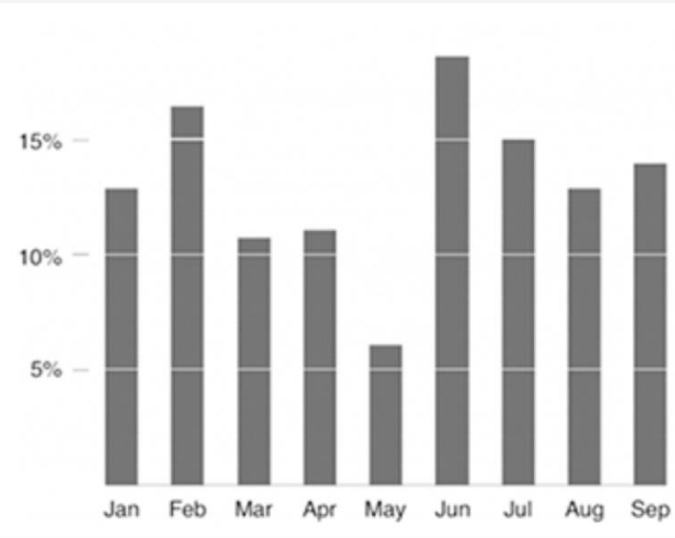
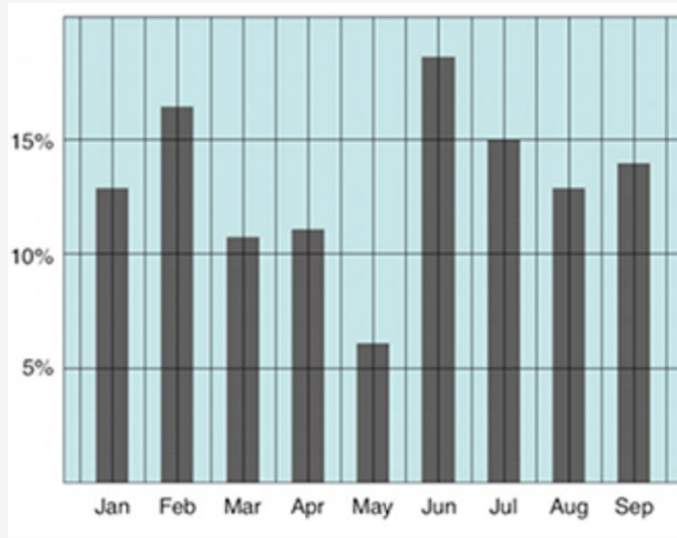
Red/green vs. blue/orange



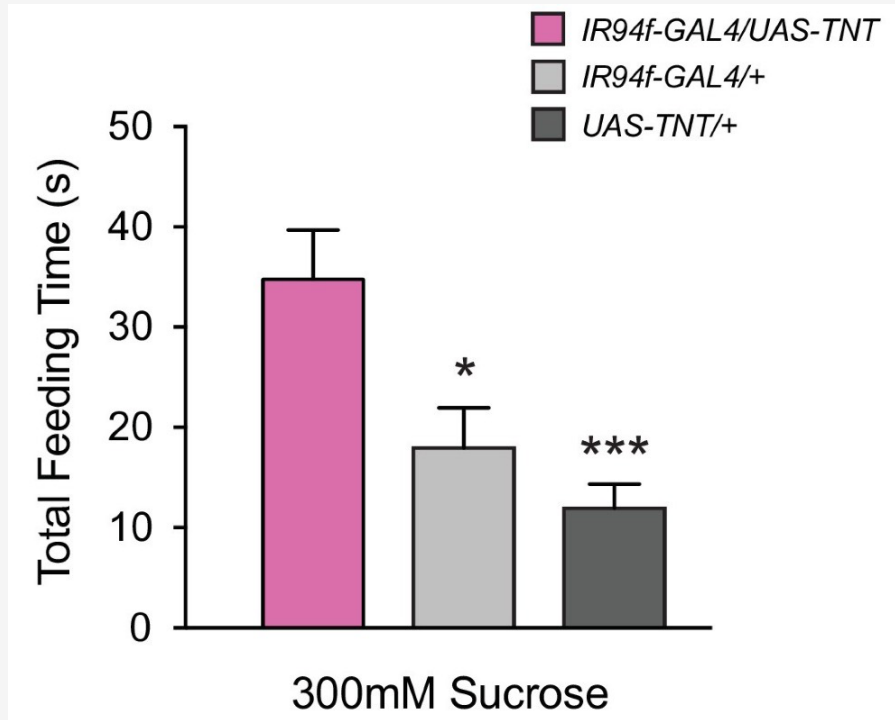
To black
and white



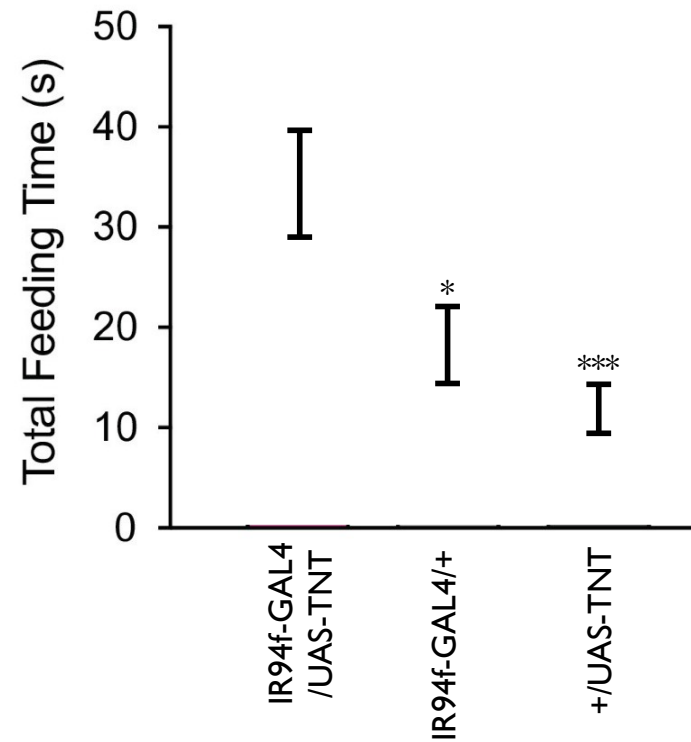
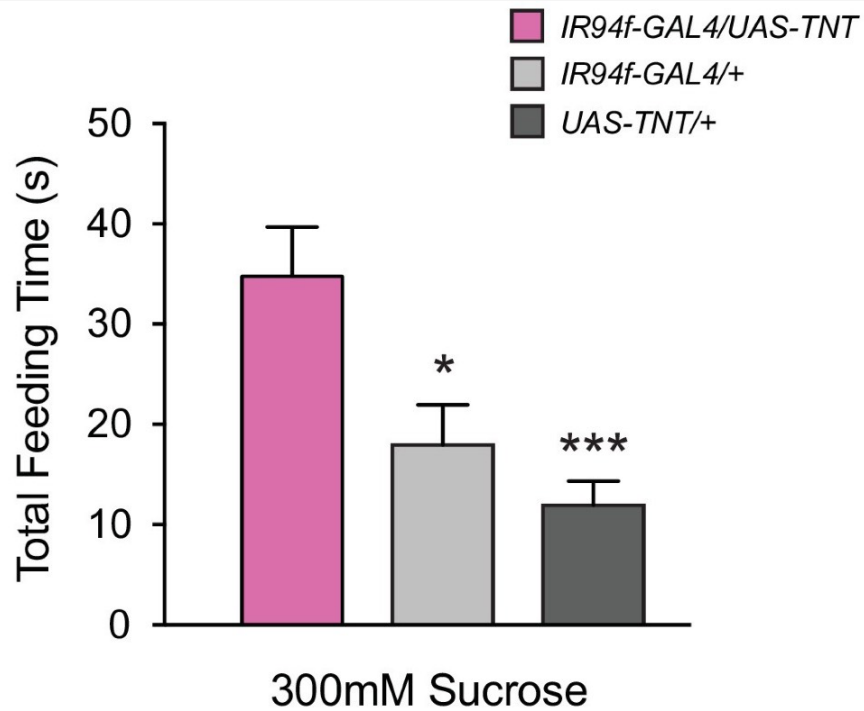
Data::Ink



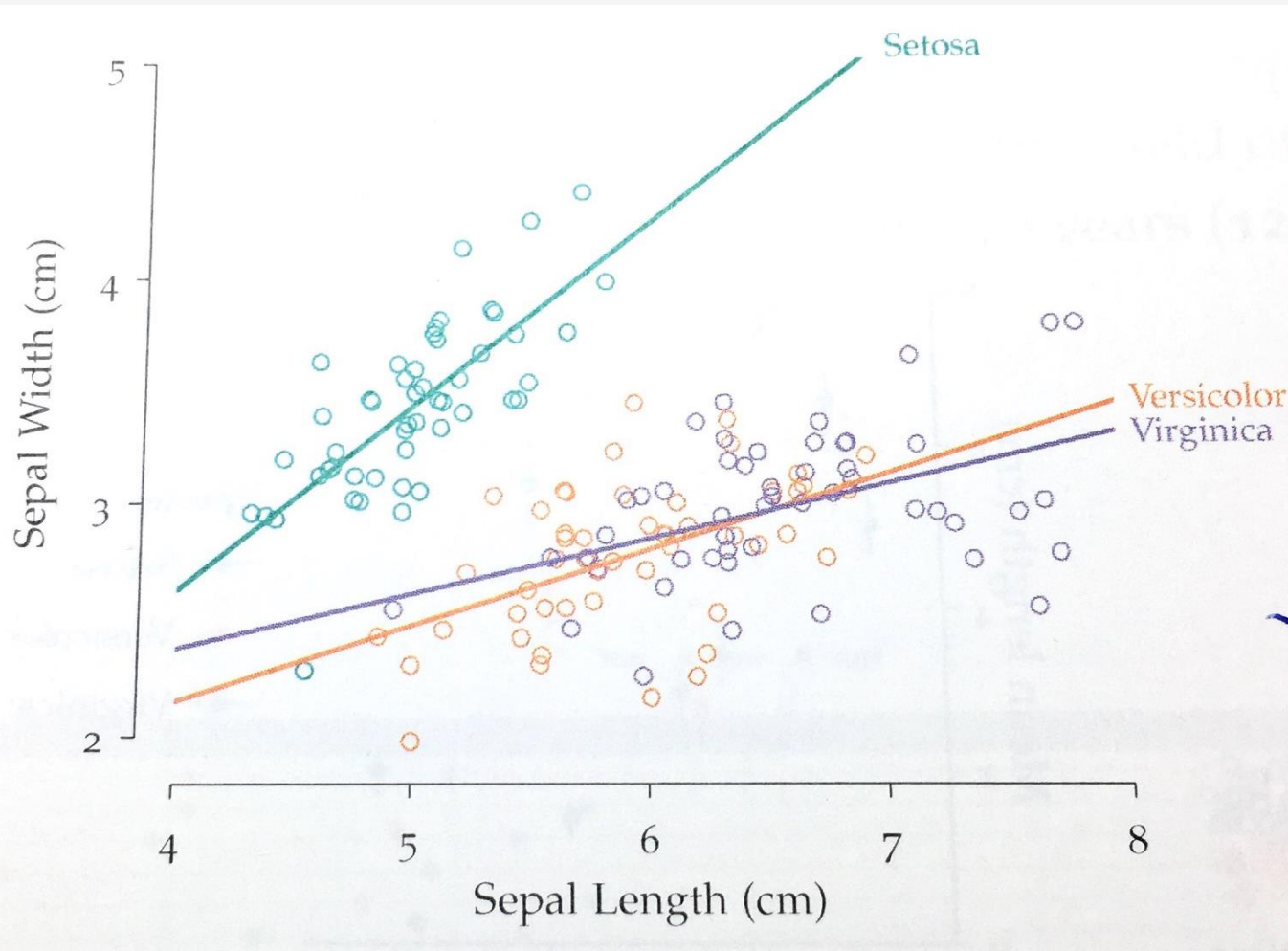
Data::Ink



Data::Ink



Data::Ink



Where is the legend?

Data::Ink

- Remove excess ink
- Show distributions, instead of bars
- Can you remove the legend?
- Remove double encodings
- Is a log scale appropriate?
- What do the 'error bars' represent?

Quote of the week

“Beyond the mountains, more mountains.”

--Haitian proverb