Week 8: *Optimization & MLE*

EMSE 6035: Marketing Analytics for Design Decisions

John Paul Helveston

October 19, 2022

Week 8: *Optimization & MLE*

1. Maximum likelihood estimation

2. Optimization (in general)

BREAK

3. Joins

4. Pilot data cleaning

Week 8: *Optimization & MLE*

1. Maximum likelihood estimation

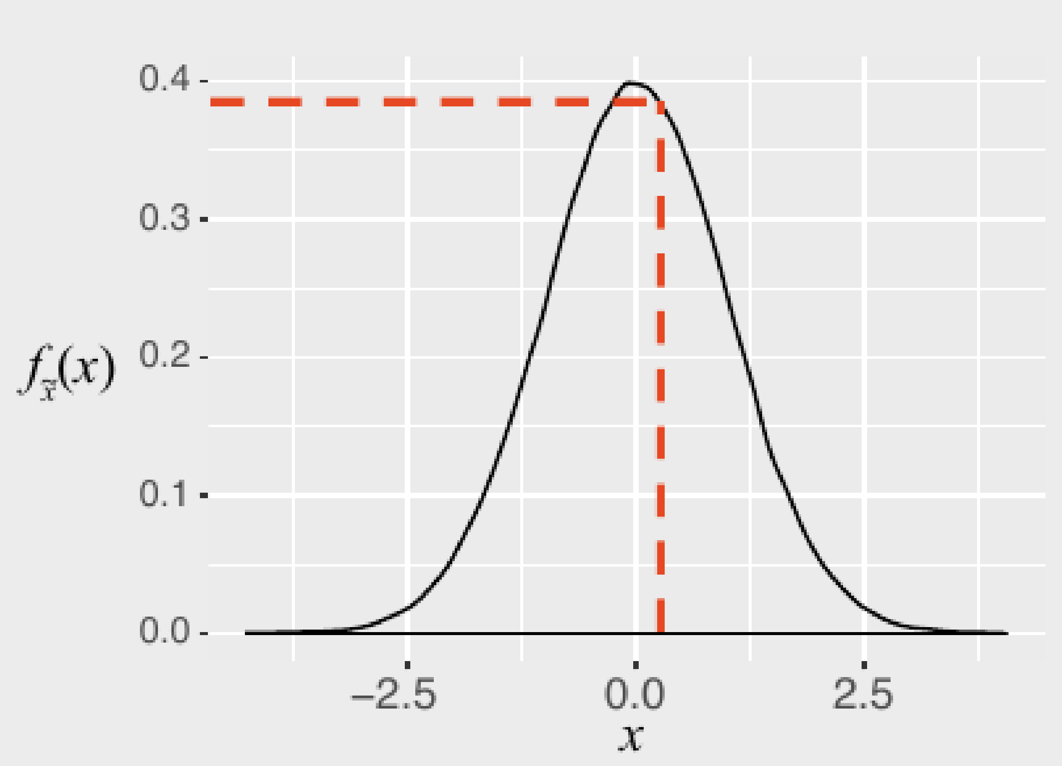
2. Optimization (in general)

BREAK

3. Joins

4. Pilot data cleaning

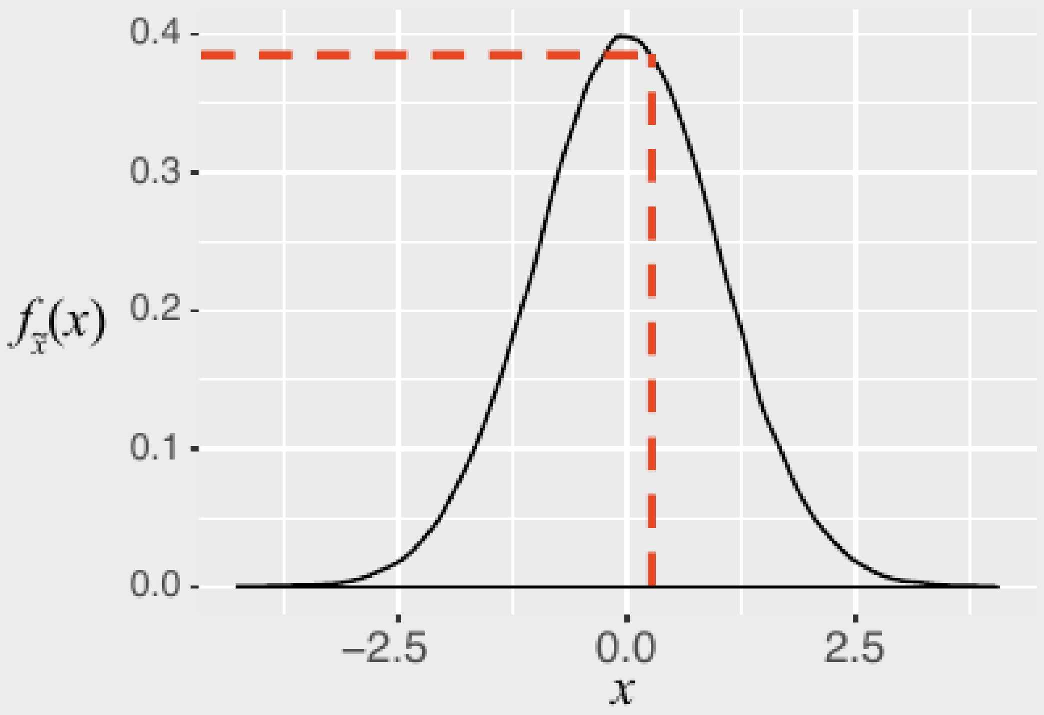
Computing the likelihood



x: an observation

$f(x)$: probability of observing x

Computing the likelihood



x: an observation

$f(x)$: probability of observing x

$\mathcal{L}(\theta|x)$: probability that θ are the true parameters, given that observed x

We want to estimate θ

We actually compute the *log*-likelihood (converts multiplication to addition)

0.39	0.35	0.24	0.39	0.40	0.11	0.33	0.35	0.07	0.37

$$\mathcal{L}(\boldsymbol{\theta}|\mathbf{x}) = f_{\tilde{x}}(x_1)\, f_{\tilde{x}}(x_2) \ldots f_{\tilde{x}}(x_n) = 1.63\text{e-}6$$

$$\log \mathcal{L}(\boldsymbol{\theta}|\mathbf{x}) = f_{\tilde{x}}(x_1) + f_{\tilde{x}}(x_2) + \ldots + f_{\tilde{x}}(x_n) = 3$$

Practice Question 1

Observations - Height of students (inches):

```
#>  [1] 65 69 66 67 68 72 68 69 63 70
```

a) Let's say we know that the height of students, $\tilde{x}$, in a classroom follows a normal distribution. A professor obtains the above height measurements students in her classroom. What is the log-likelihood that $\tilde{x} \sim \mathcal{N}(68, 4)$? In other words, compute $\ln \mathcal{L}(\mu = 68, \sigma = 4)$.

b) Compute the log-likelihood function using the same standard deviation $(\sigma = 4)$ but with the following different values for the mean, $\mu : 66, 67, 68, 69, 70$. How do the results compare? Which value for μ produces the highest log-likelihood?

Week 8: *Optimization & MLE*

1. Maximum likelihood estimation

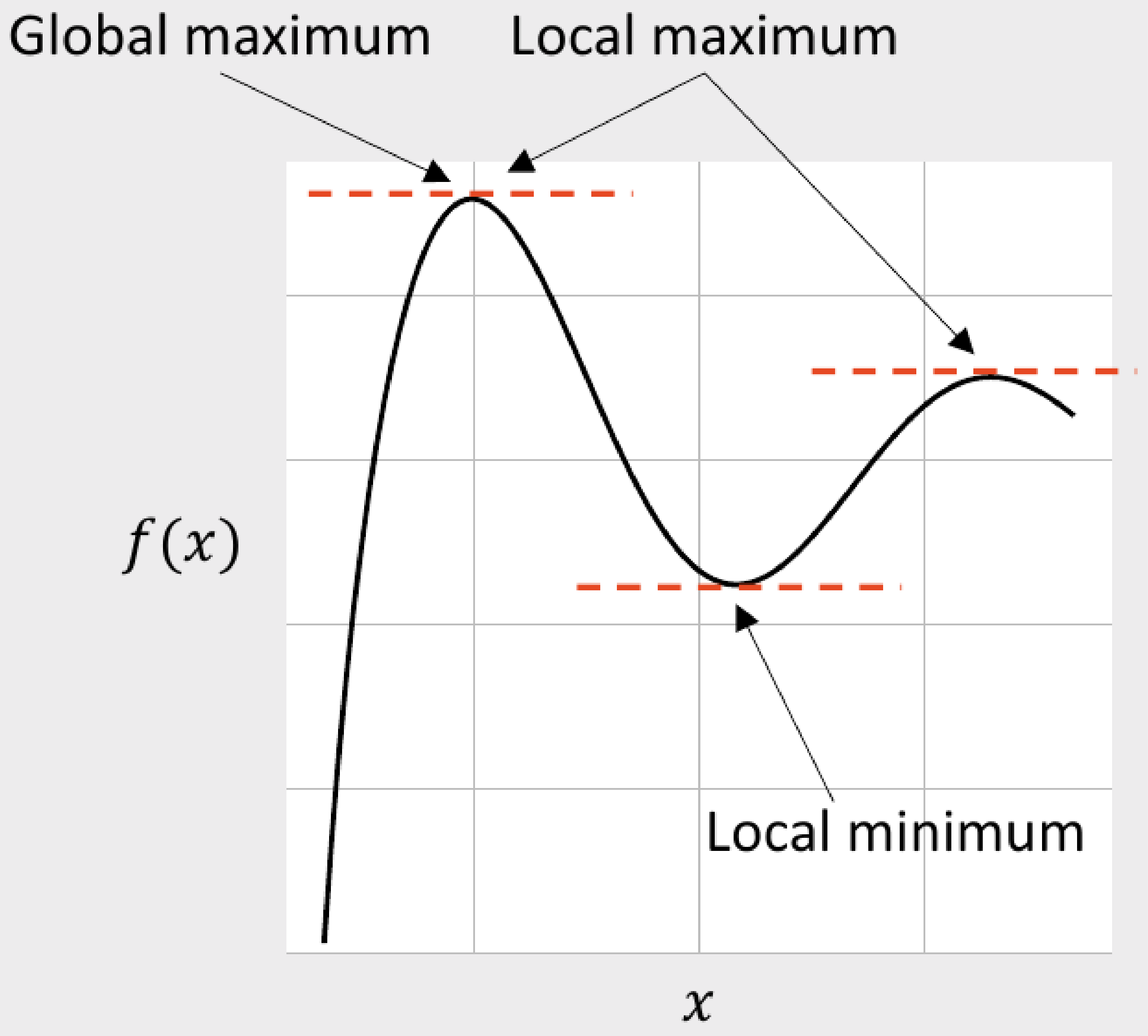
2. Optimization (in general)

BREAK

3. Joins

4. Pilot data cleaning

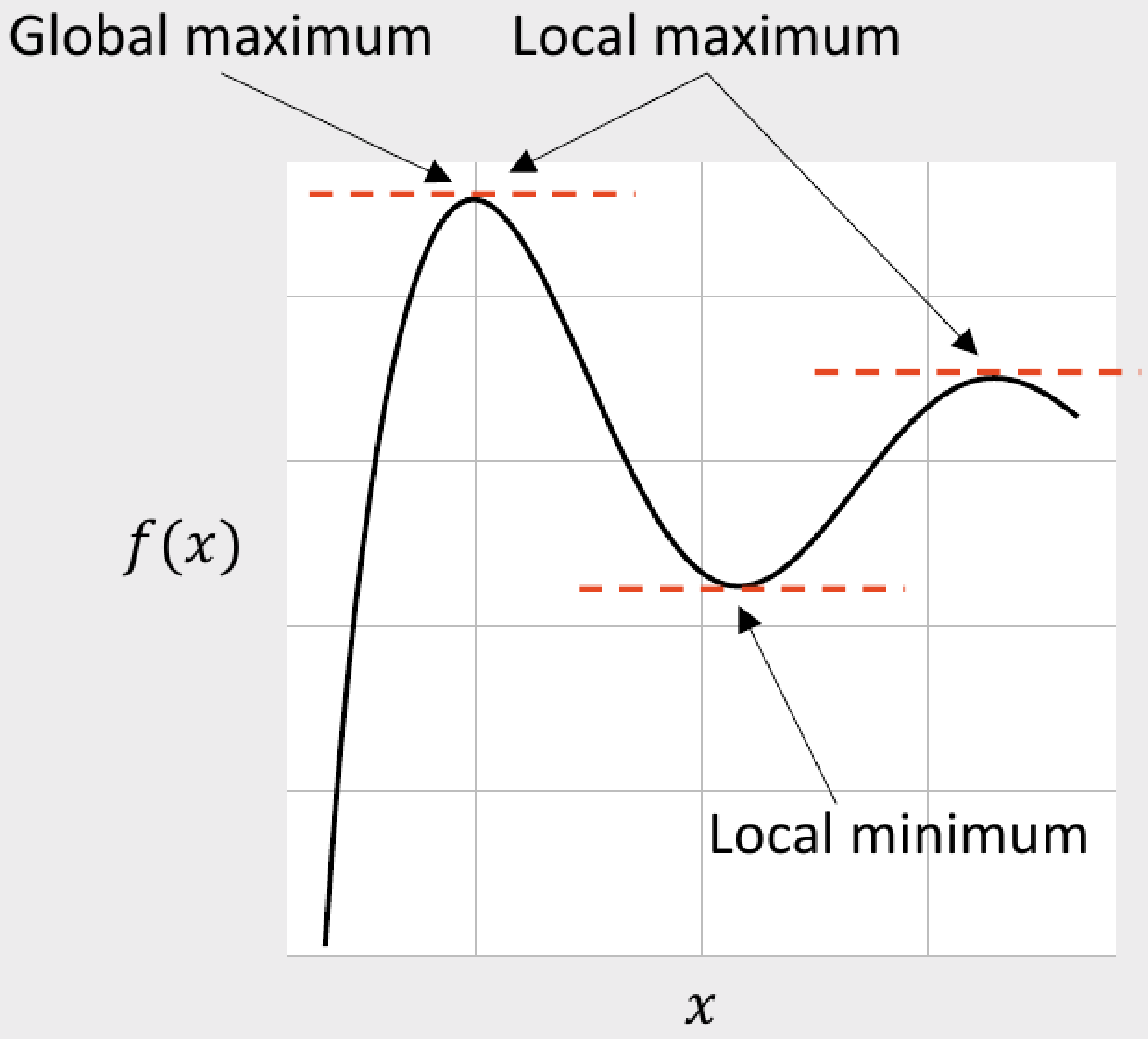
$$f(x)$$



First order necessary condition

x^* is a "stationary point" when

$$\frac{df(x^*)}{dx} = 0$$



First order necessary condition

x^* is a "stationary point" when

$$\frac{df(x^*)}{dx} = 0$$

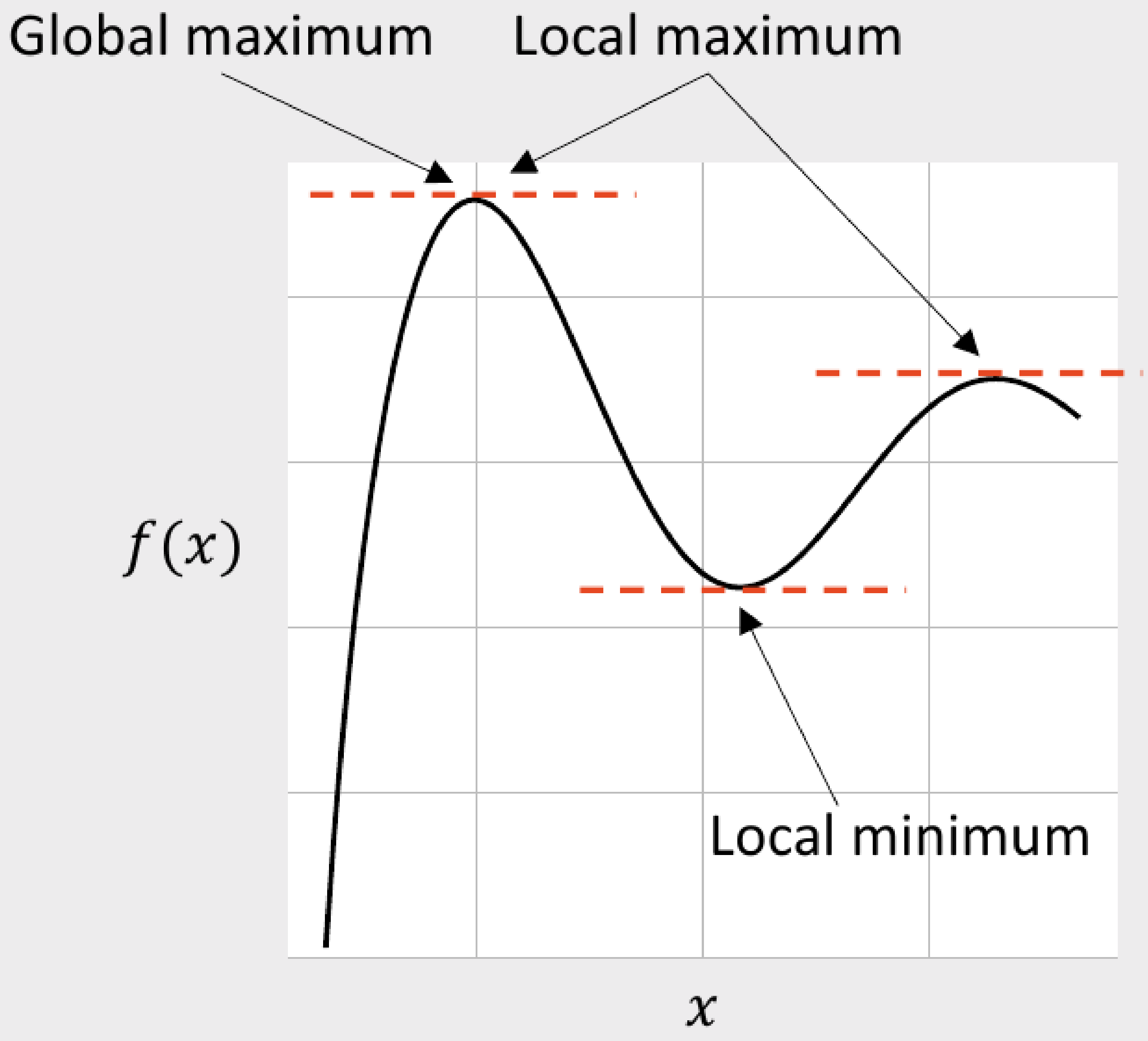
Second order sufficiency condition

x^* is a local *maximum* when

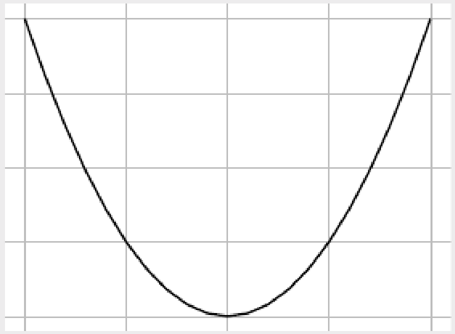
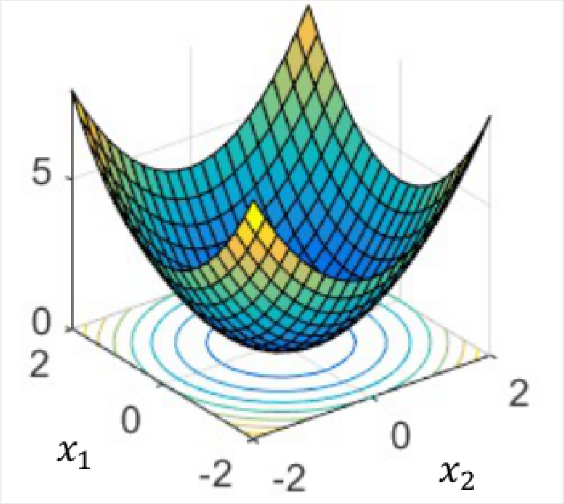
$$\frac{d^2 f(x^*)}{dx^2} < 0$$

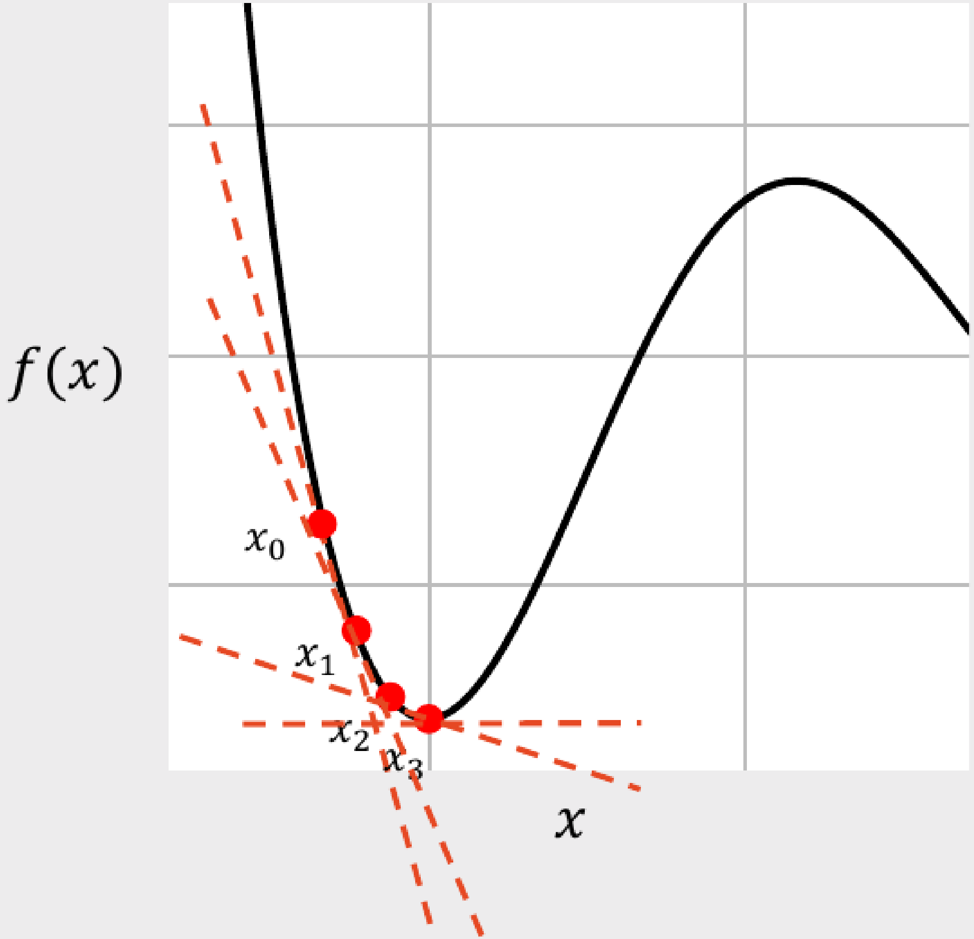
x^* is a local *minimum* when

$$\frac{d^2 f(x^*)}{dx^2} > 0$$



Optimality conditions for local **minimum**

Number of dimensions	First order condition	Second order condition	Example
One	$\frac{df(x^*)}{dx} = 0$	$\frac{d^2f(x^*)}{dx^2} > 0$	
Multiple	"Gradient" $\nabla f(x_1, x_2, \ldots x_n)$ $= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]$ $= [0,0, \ldots, 0]$	"Hessian" $\nabla^2 f(x_1, x_2, \ldots x_n)$ $= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$ Must be "positive definite"	



<u>Gradient Descent Method</u>:

1. Choose a starting point, x_0
2. At that point, compute the gradient, $\nabla f(x_0)$
3. Compute the next point, with a step size γ :

$$x_{n+1} = x_n - \gamma \nabla f(x_n)$$

Very small number

*Stop when $\nabla f(x_n) < \delta$

or

*Stop when $(x_{n+1} - x_n) < \delta$

Practice Question 2

Consider the following function:

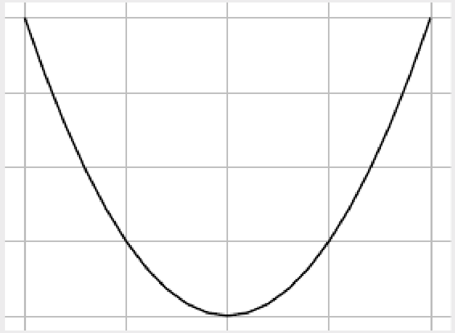
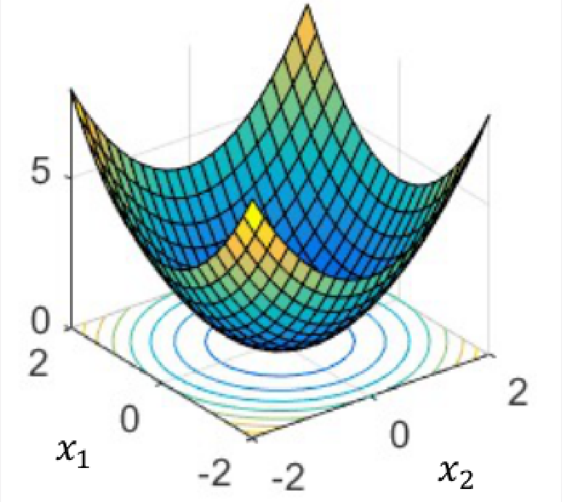
$$f(x) = x^2 - 6x$$

The gradient is:

$$\nabla f(x) = 2x - 6$$

Using the starting point $x = 1$ and the step size $\gamma = 0.3$, apply the gradient descent method to compute the next **three** points in the search algorithm.

Optimality conditions for local **minimum**

Number of dimensions	First order condition	Second order condition	Example
One	$\frac{df(x^*)}{dx} = 0$	$\frac{d^2f(x^*)}{dx^2} > 0$	
Multiple	"Gradient" $\nabla f(x_1, x_2, \ldots x_n)$ $= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]$ $= [0,0, \ldots, 0]$	"Hessian" $\nabla^2 f(x_1, x_2, \ldots x_n)$ $= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$ Must be "positive definite"	

Practice Question 3

Consider the following function:

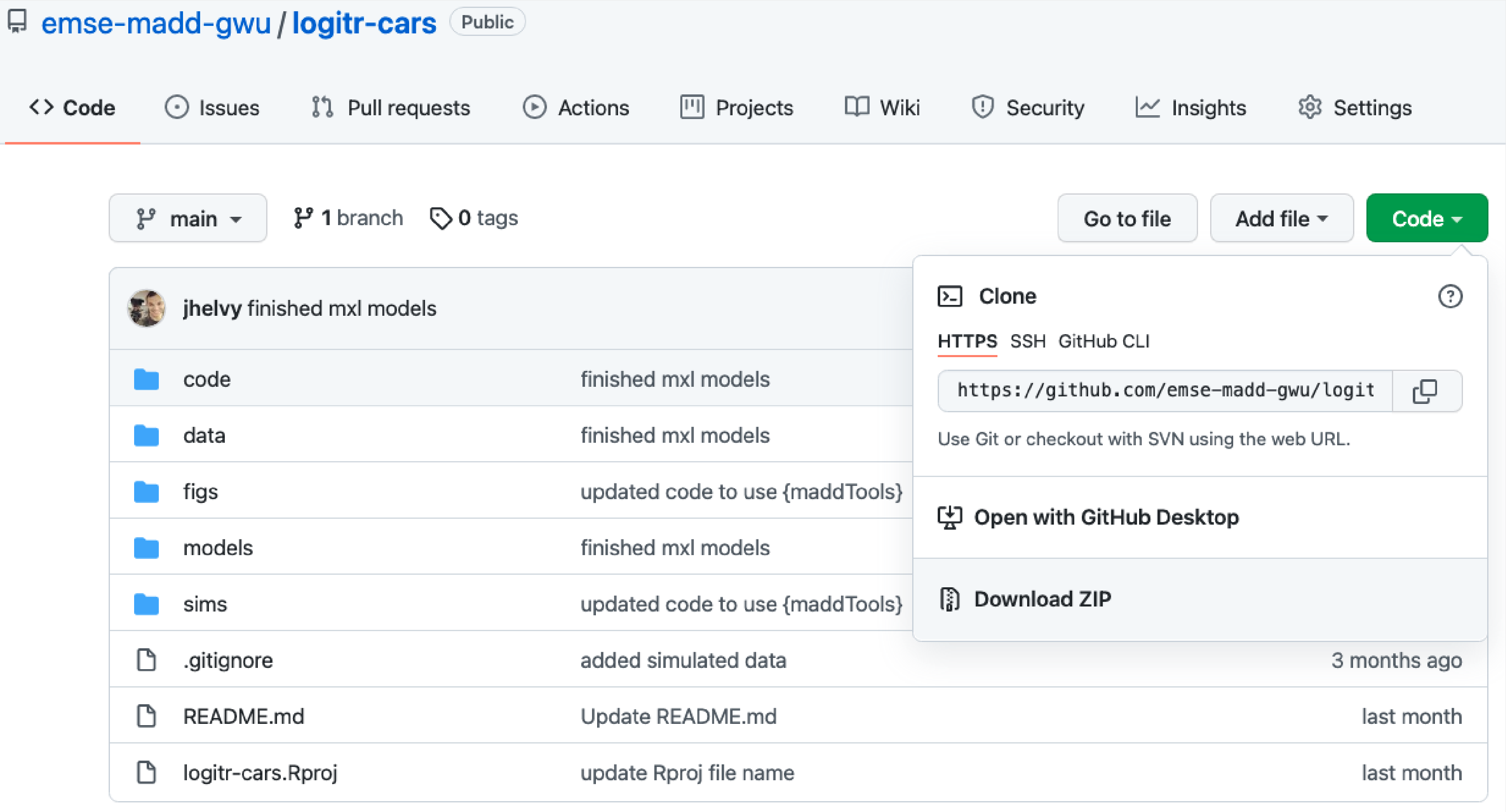
$$f(\underline{x}) = x_1^2 + 4x_2^2$$

The gradient is:

$$\nabla f(\underline{x}) = \begin{bmatrix} 2x_1 \\ 8x_2 \end{bmatrix}$$

Using the starting point $\underline{x}_0 = [1, 1]$ and the step size $\gamma = 0.15$, apply the gradient descent method to compute the next **three** points in the search algorithm.

Download the logitr-cars repo from GitHub



Estimating utility models

1. Open `logitr-cars.Rproj`
2. Open `code/3.1-model-mnl.R`

Maximum likelihood estimation

$$\tilde{u}_j = v_j + \tilde{\varepsilon}_j$$
$$= \beta_1 x_{j1} + \beta_2 x_{j2} + \dots + \tilde{\varepsilon}_j$$
$$= \boldsymbol{\beta}' \mathbf{x}_j + \tilde{\varepsilon}_j$$

Estimate $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_n]$
by maximizing the likelihood function

$$\boxed{\text{minimize } -log\mathcal{L} = -\sum_{j=1}^{J} P_j(\boldsymbol{\beta}|\mathbf{x})^{y_j} \quad \text{with respect to } \boldsymbol{\beta}}$$

$y_j = 1$ if alternative j was chosen
$y_j = 0$ if alternative j was not chosen

For logit model:

$$P_j = \frac{e^{v_j}}{\sum_{k=1}^{J} e^{v_k}} = \frac{e^{\boldsymbol{\beta}' \mathbf{x}_j}}{\sum_{k=1}^{J} e^{\boldsymbol{\beta}' \mathbf{x}_k}}$$

Break

05:00

Week 8: *Optimization & MLE*

1. Maximum likelihood estimation

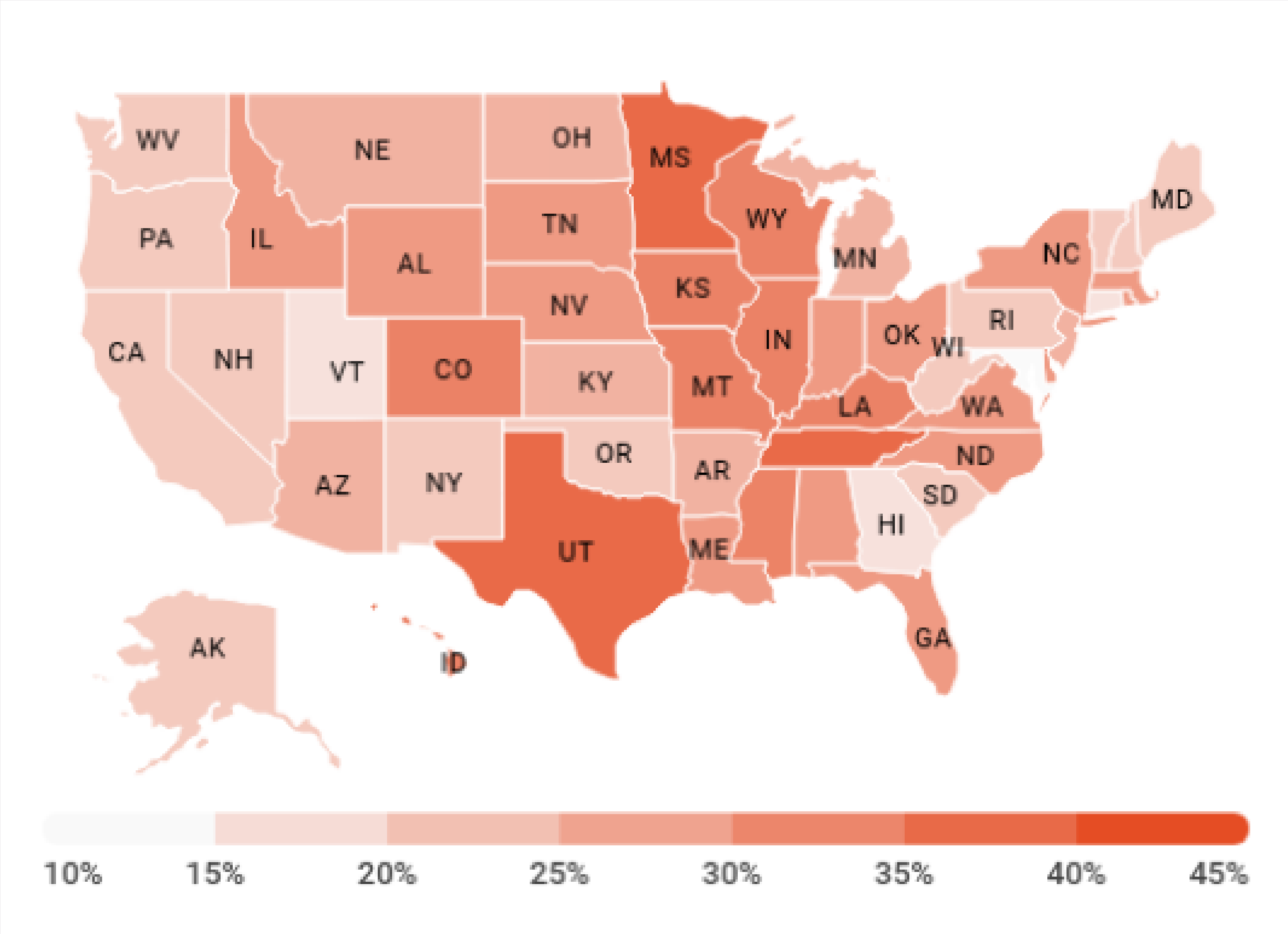
2. Optimization (in general)

BREAK

3. Joins

4. Pilot data cleaning

What's wrong with this map?



Likely culprit: Merging two columns

```r
head(names)
```

```
#>                 state_name
#> 1                  Alabama
#> 2                   Alaska
#> 3                  Arizona
#> 4                 Arkansas
#> 5   Armed Forces Africa
#> 6 Armed Forces Americas
```

```r
head(abbs)
```

```
#>   state_abb
#> 1        AA
#> 2        AE
#> 3        AE
#> 4        AE
#> 5        AE
#> 6        AK
```

```r
result <- cbind(names, abbs)
head(result)
```

```
#>                 state_name state_abb
#> 1                  Alabama        AA
#> 2                   Alaska        AE
#> 3                  Arizona        AE
#> 4                 Arkansas        AE
#> 5   Armed Forces Africa        AE
#> 6 Armed Forces Americas        AK
```

Joins

1. `inner_join()`
2. `left_join()` / `right_join()`
3. `full_join()`

Example: `band_members` & `band_instruments`

`band_members`

```
#> # A tibble: 3 × 2
#>   name  band
#>   <chr> <chr>
#> 1 Mick  Stones
#> 2 John  Beatles
#> 3 Paul  Beatles
```

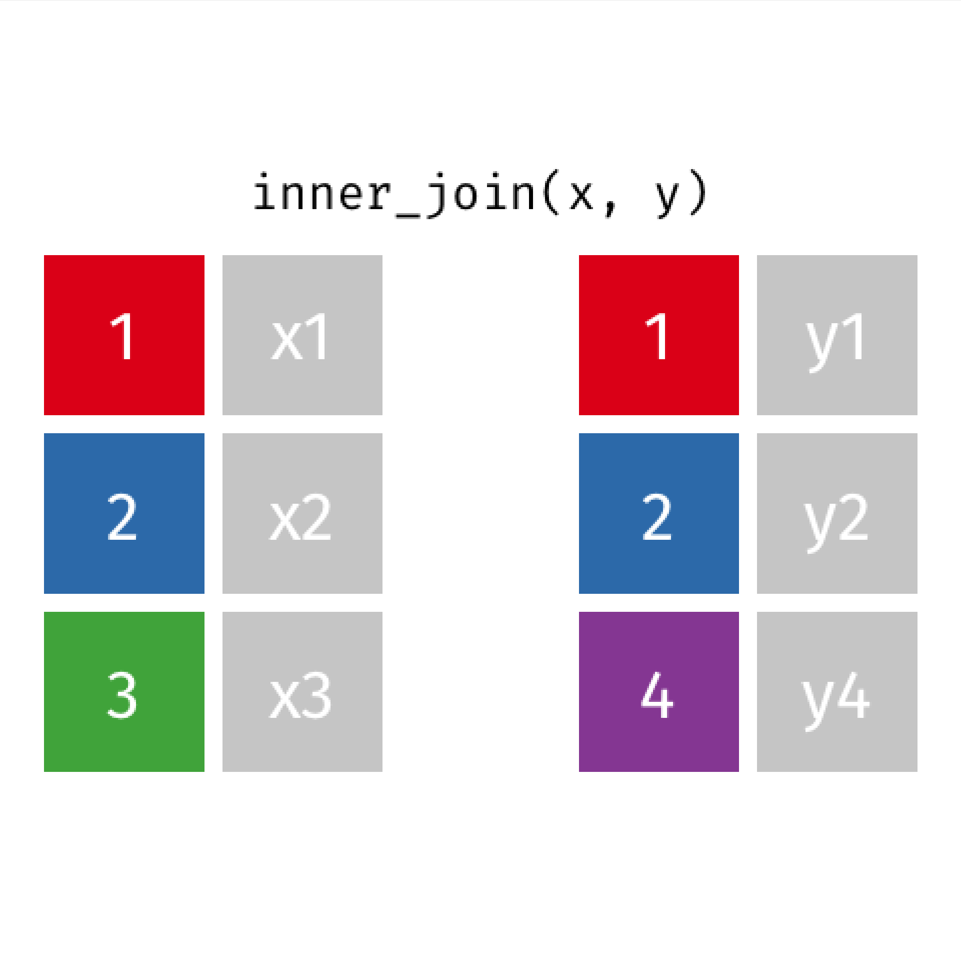
`band_instruments`

```
#> # A tibble: 3 × 2
#>   name  plays
#>   <chr> <chr>
#> 1 John  guitar
#> 2 Paul  bass
#> 3 Keith guitar
```

inner_join()

```
band_members %>%
    inner_join(band_instruments)
```

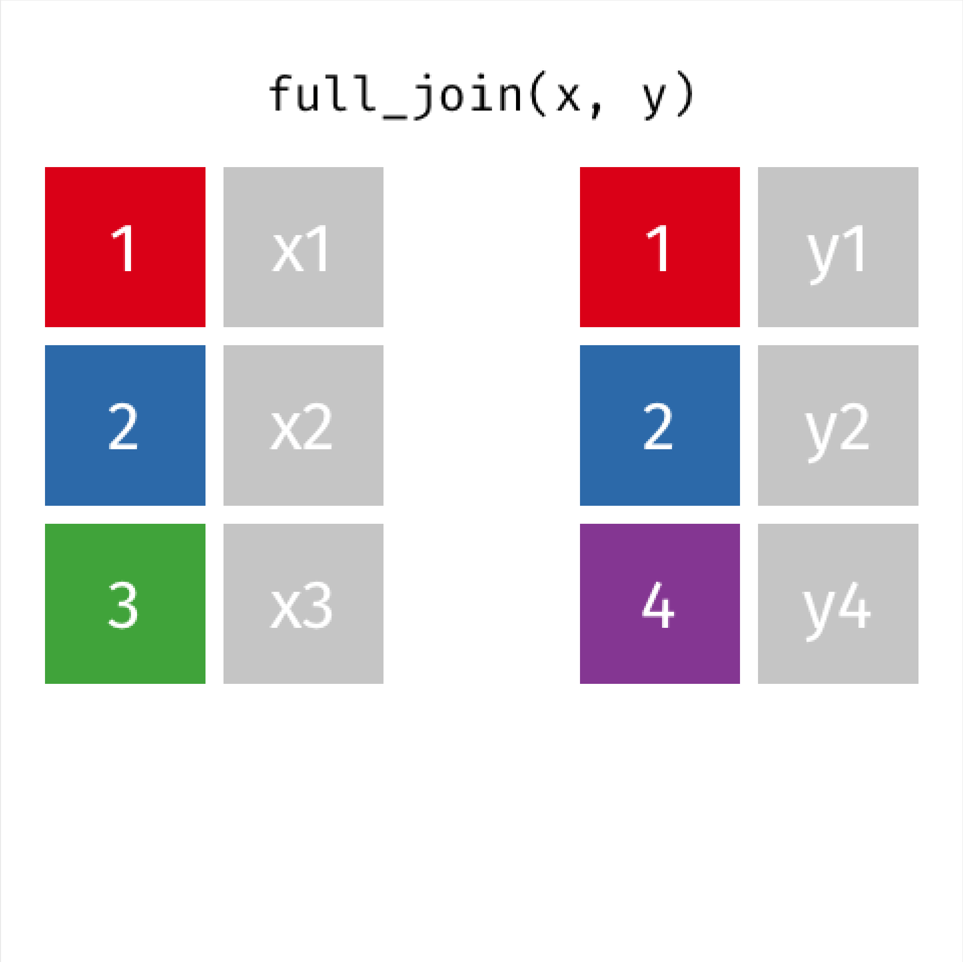
```
#> # A tibble: 2 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 John  Beatles guitar
#> 2 Paul  Beatles bass
```



full_join()

```
band_members %>%
    full_join(band_instruments)
```

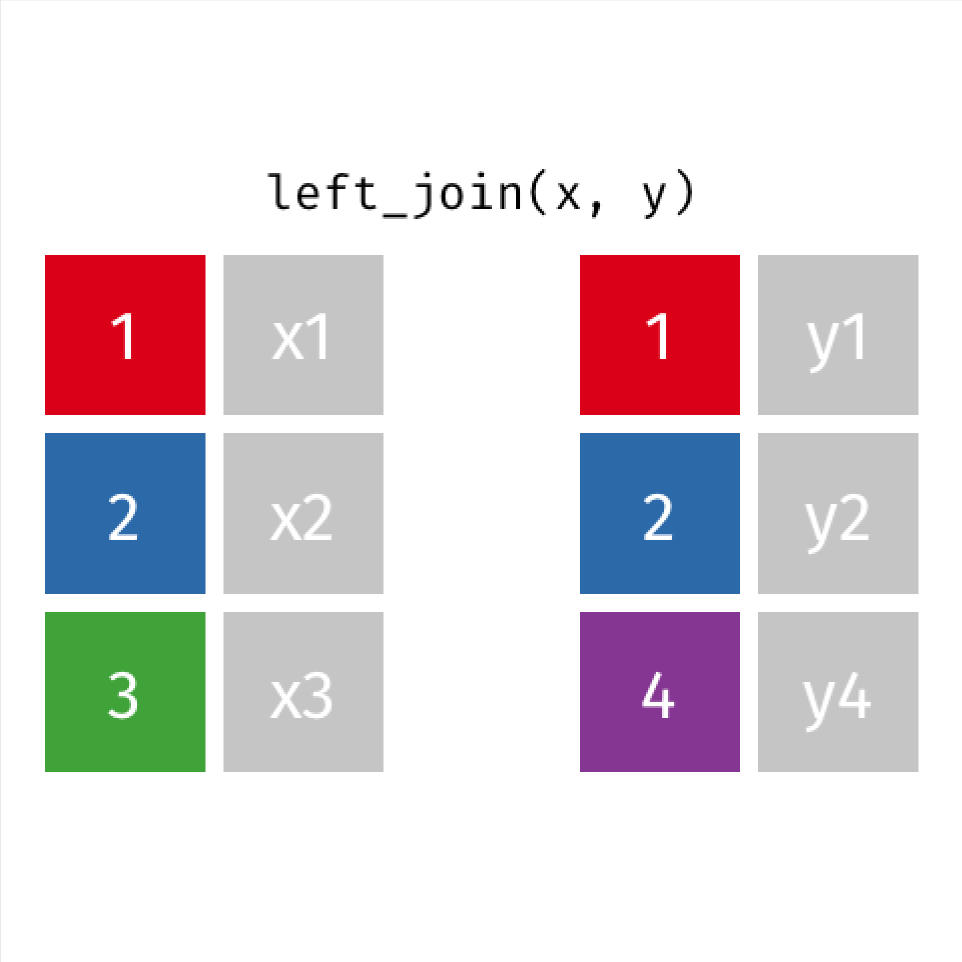
```
#> # A tibble: 4 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 Mick  Stones  <NA>
#> 2 John  Beatles guitar
#> 3 Paul  Beatles bass
#> 4 Keith <NA>    guitar
```



left_join()

```r
band_members %>%
    left_join(band_instruments)
```

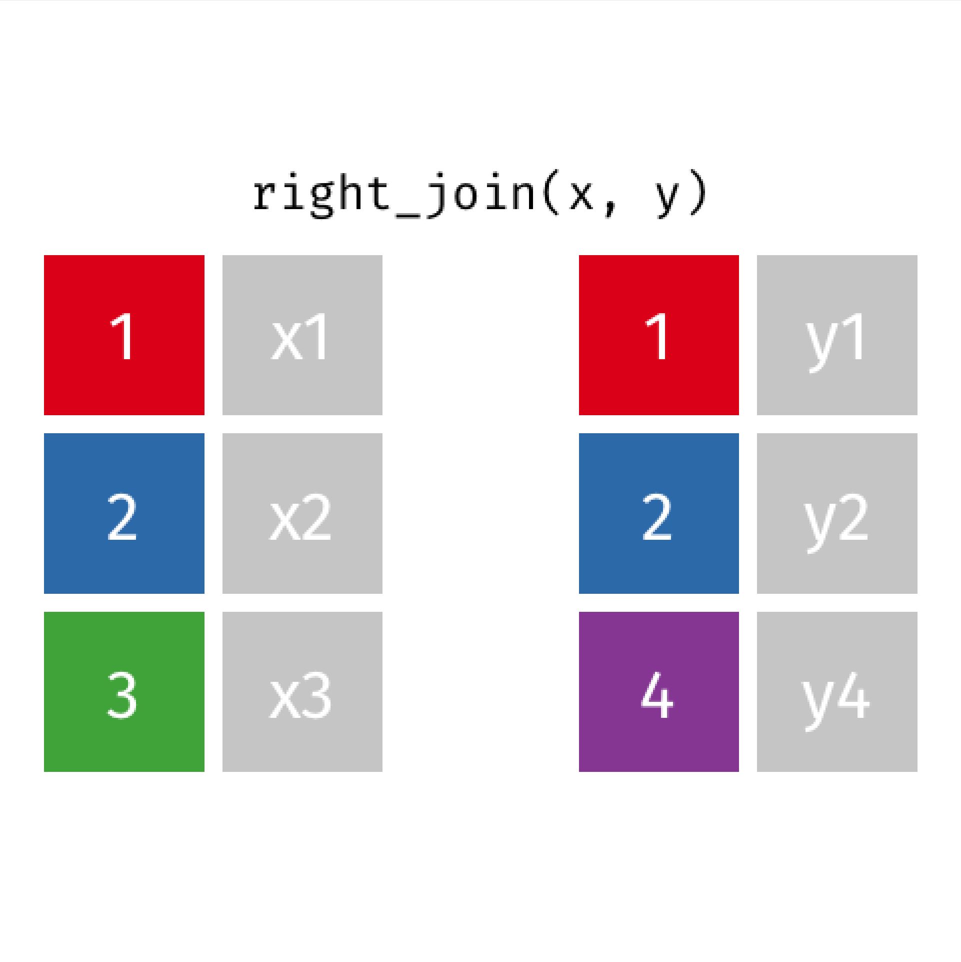
```
#> # A tibble: 3 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 Mick  Stones  <NA>
#> 2 John  Beatles guitar
#> 3 Paul  Beatles bass
```



right_join()

```
band_members %>%
    right_join(band_instruments)
```

```
#> # A tibble: 3 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 John  Beatles guitar
#> 2 Paul  Beatles bass
#> 3 Keith <NA>    guitar
```



Specify the joining variable name

```r
band_members %>%
    left_join(band_instruments)
```

```
#> Joining, by = "name"
```

```
#> # A tibble: 3 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 Mick  Stones  <NA>
#> 2 John  Beatles guitar
#> 3 Paul  Beatles bass
```

```r
band_members %>%
    left_join(band_instruments,
              by = 'name')
```

```
#> # A tibble: 3 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 Mick  Stones  <NA>
#> 2 John  Beatles guitar
#> 3 Paul  Beatles bass
```

Specify the joining variable name

If the names differ, use `by = c("left_name" = "joining_name")`

```
band_members
```

```
#> # A tibble: 3 × 2
#>   name  band
#>   <chr> <chr>
#> 1 Mick  Stones
#> 2 John  Beatles
#> 3 Paul  Beatles
```

```
band_instruments2
```

```
#> # A tibble: 3 × 2
#>   artist plays
#>   <chr>  <chr>
#> 1 John   guitar
#> 2 Paul   bass
#> 3 Keith  guitar
```

```
band_members %>%
    left_join(band_instruments2,
              by = c("name" = "artist"))
```

```
#> # A tibble: 3 × 3
#>   name  band    plays
#>   <chr> <chr>   <chr>
#> 1 Mick  Stones  <NA>
#> 2 John  Beatles guitar
#> 3 Paul  Beatles bass
```

Specify the joining variable name

Or just rename the joining variable in a pipe

```
band_members
```

```
#> # A tibble: 3 × 2
#>   name  band
#>   <chr> <chr>
#> 1 Mick  Stones
#> 2 John  Beatles
#> 3 Paul  Beatles
```

```
band_instruments2
```

```
#> # A tibble: 3 × 2
#>   artist plays
#>   <chr>  <chr>
#> 1 John   guitar
#> 2 Paul   bass
#> 3 Keith  guitar
```

```
band_members %>%
    rename(artist = name) %>%
    left_join(band_instruments2,
              by = "artist")
```

```
#> # A tibble: 3 × 3
#>   artist band    plays
#>   <chr>  <chr>   <chr>
#> 1 Mick   Stones  <NA>
#> 2 John   Beatles guitar
#> 3 Paul   Beatles bass
```

Your turn

15:00

1. Write code to read in the `state_abbs.csv` and `state_regions.csv` data files in the "data" folder.
2. Create a new data frame called `states` by joining the two data frames `states_abbs` and `state_regions` together. The result should be a data frame with variables `region`, `name`, `abb`.

Your result should look like this:

```
head(states)
```

```
#> # A tibble: 6 × 3
#>   region    name          abb
#>   <chr>     <chr>         <chr>
#> 1 Northeast Maine         ME
#> 2 Northeast New Hampshire NH
#> 3 Northeast Vermont       VT
#> 4 Northeast Massachusetts MA
#> 5 Northeast Rhode Island  RI
#> 6 Northeast Connecticut   CT
```

Week 8: *Optimization & MLE*

1. Maximum likelihood estimation

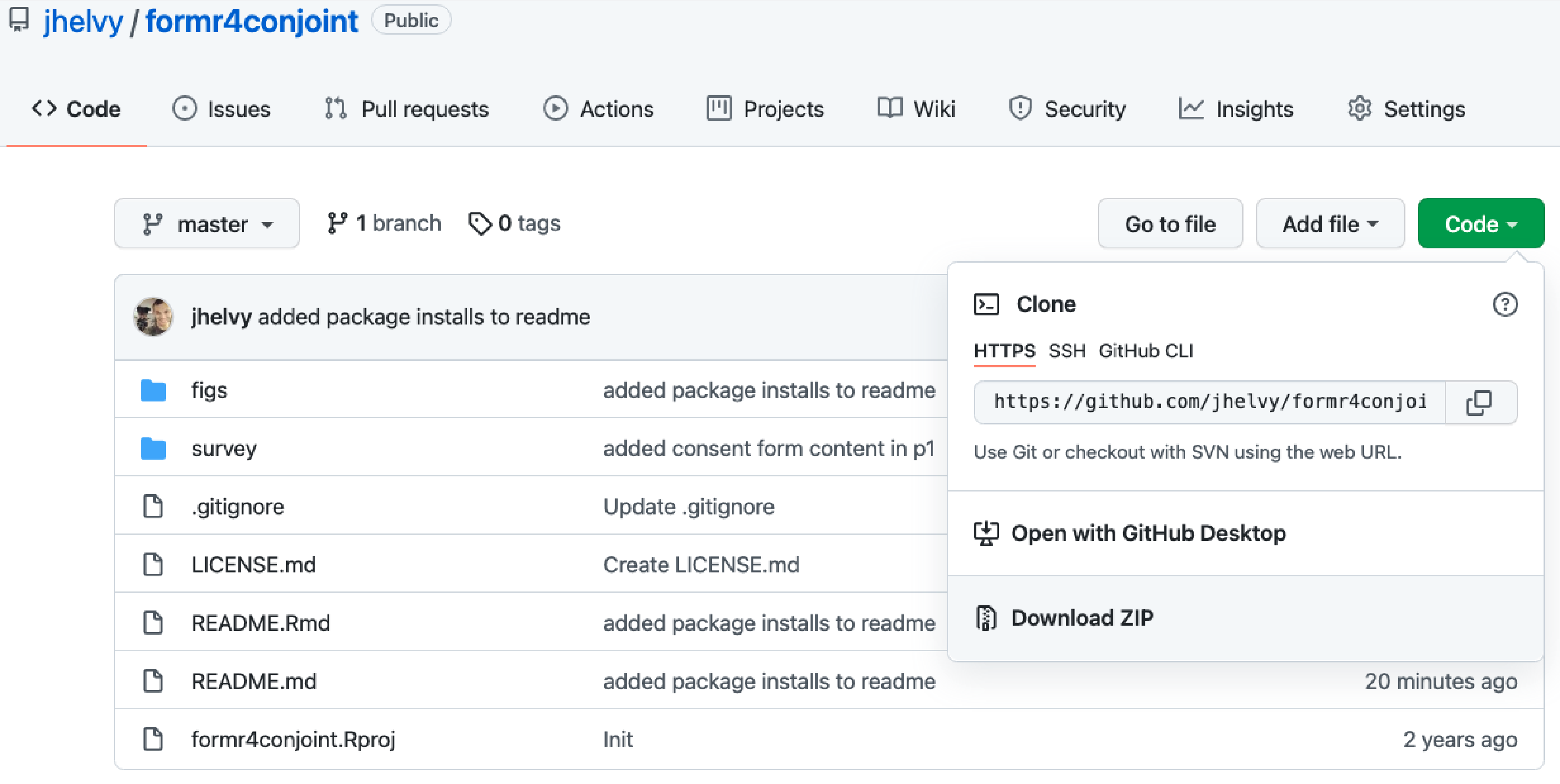
2. Optimization (in general)

BREAK

3. Joins

4. Pilot data cleaning

Download the formr4conjoint repo from GitHub



Cleaning formr survey data

1. Open `formr4conjoint.Rproj`
2. Open `code/data_cleaning.R`

Team time

For the rest of class, work with your team mates to start importing and cleaning your pilot survey data