



Week 8: *Optimization & MLE*

 EMSE 6035: Marketing Analytics for Design Decisions

 John Paul Helveston

 October 18, 2023

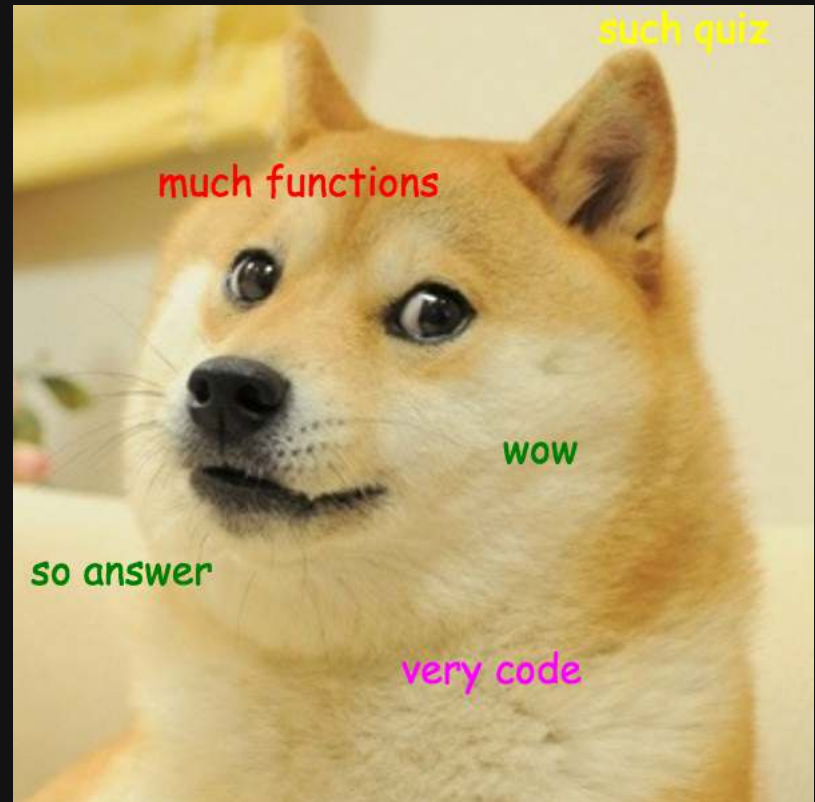
Quiz 3

Download the template from the #class channel

Make sure you unzip it!

When done, submit your `quiz3.qmd` on Blackboard

10:00



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

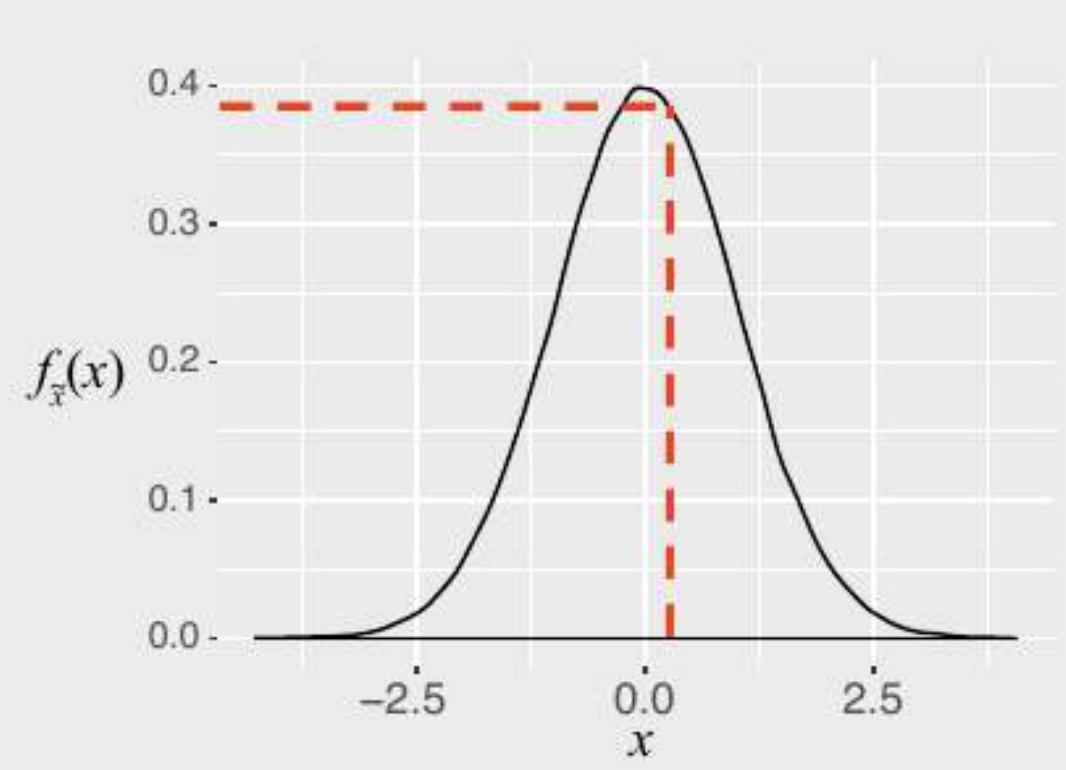
2. Optimization (in general)

BREAK

3. Joins

4. Pilot data cleaning

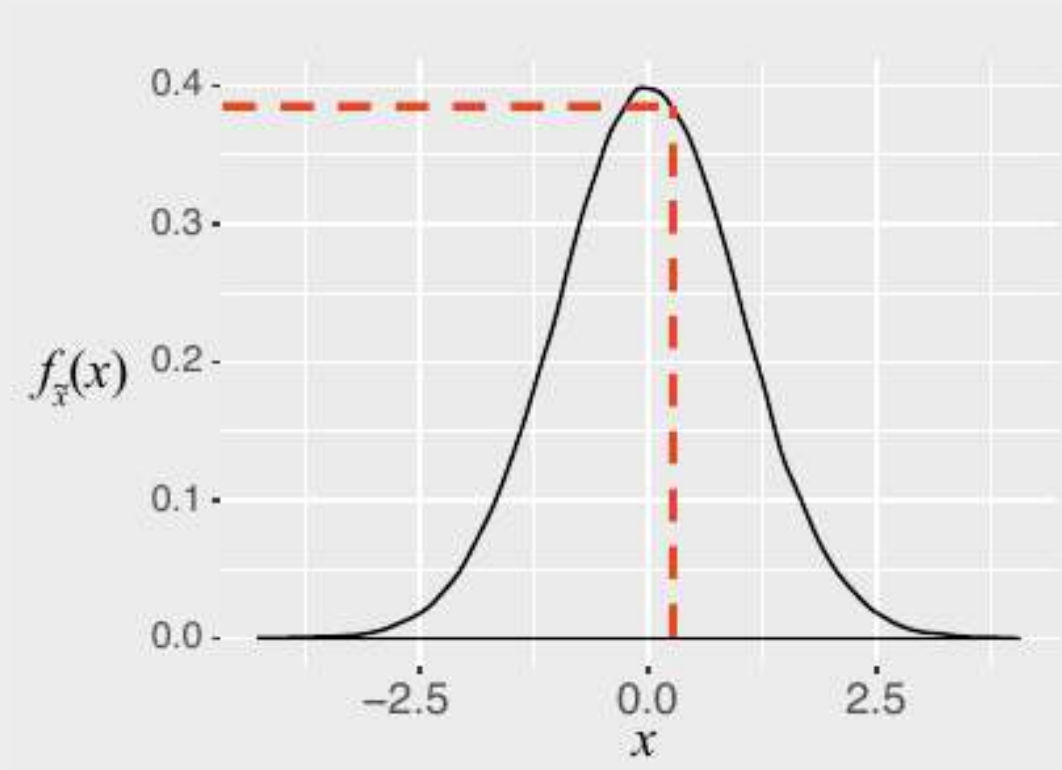
Computing the likelihood



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
------	------	------	------	------	------	------	------	------	------

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

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

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, μ : 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

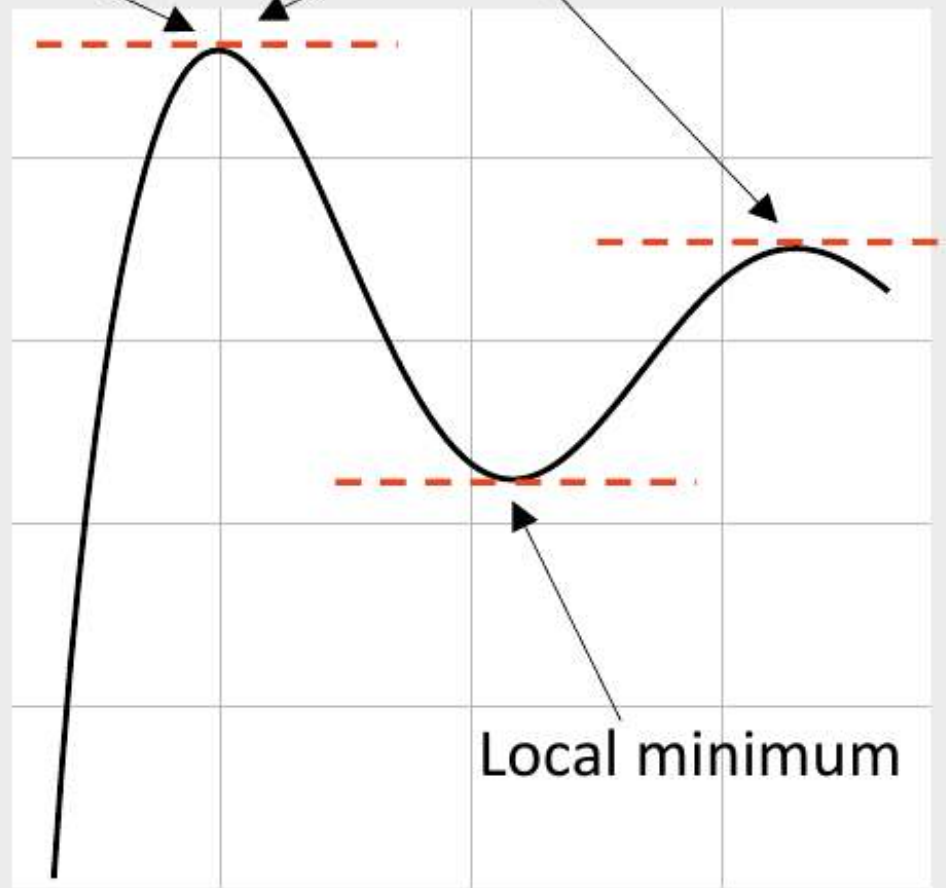
4. Pilot data cleaning

$$f(x)$$

Global maximum

Local maximum

$f(x)$



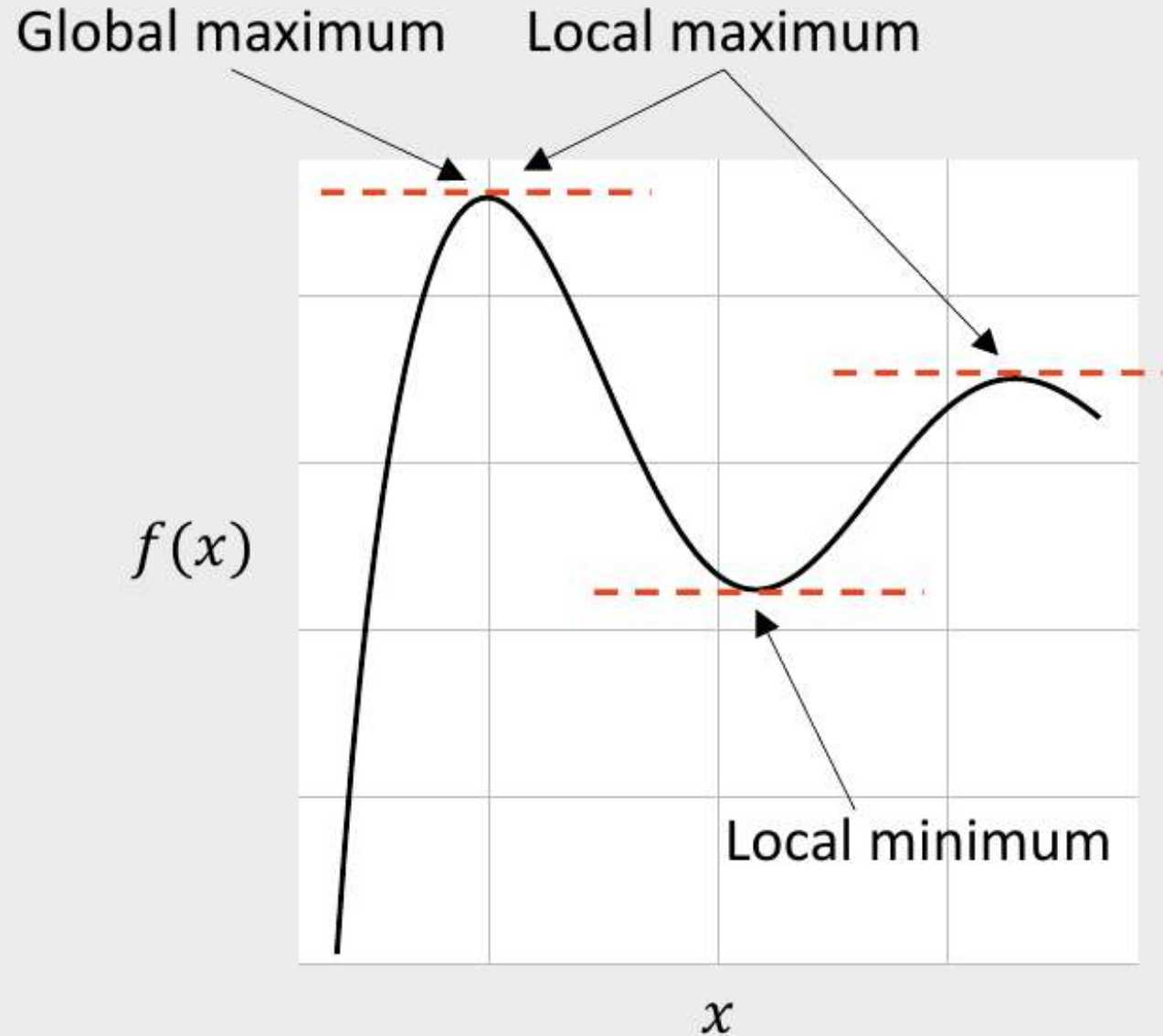
Local minimum

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

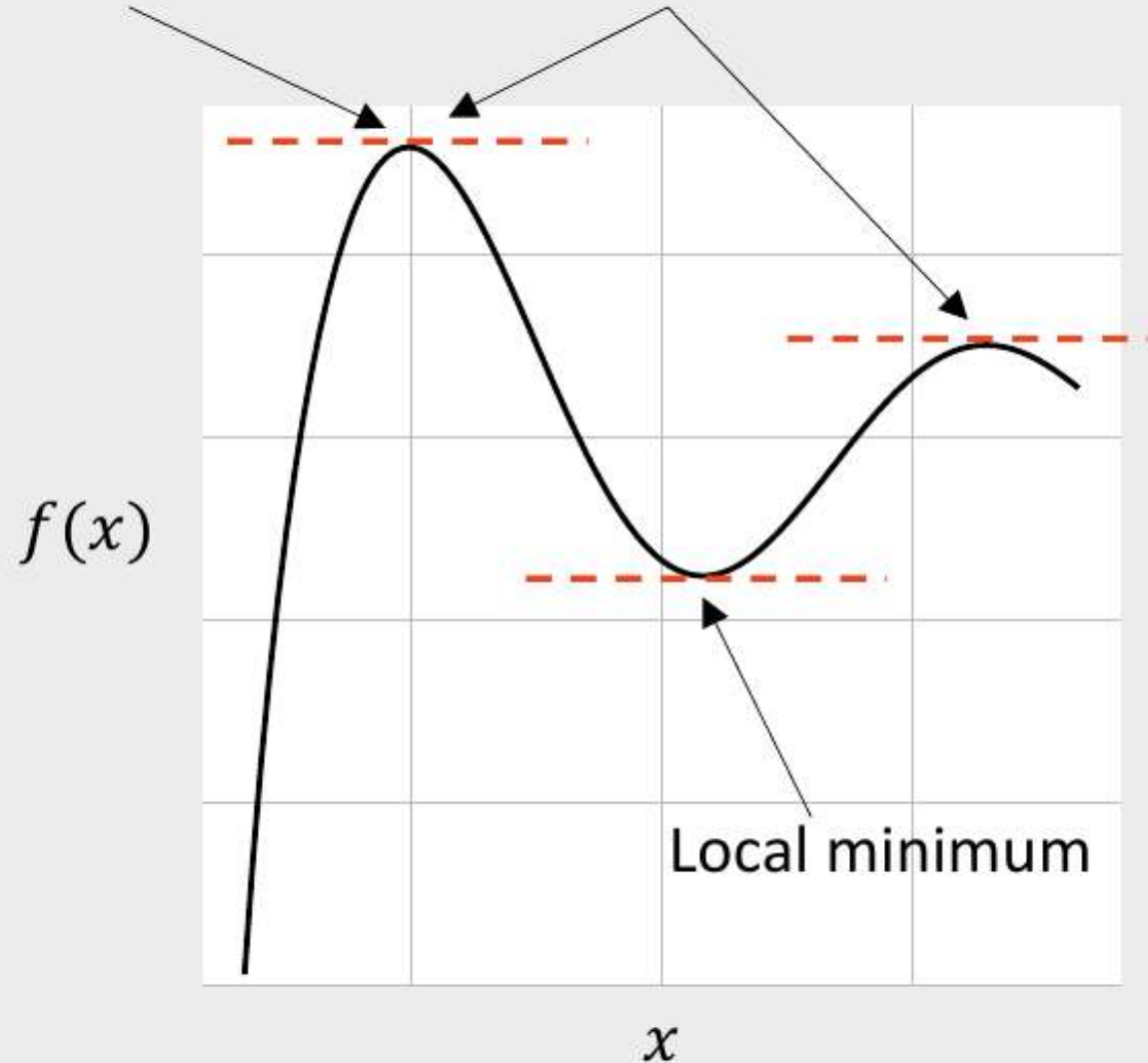
x^* is a local *maximum* when

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

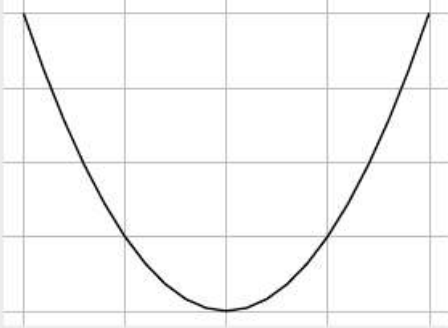
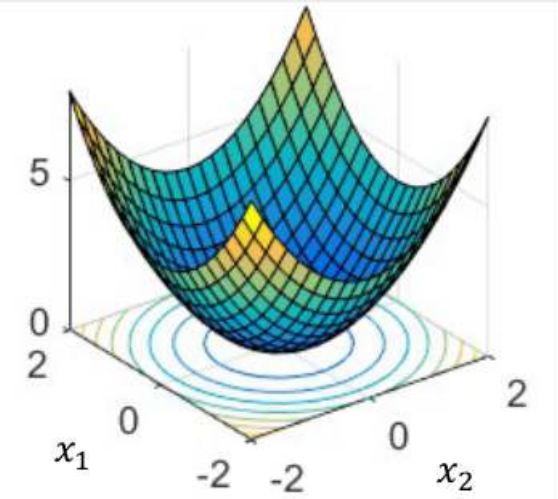
x^* is a local *minimum* when

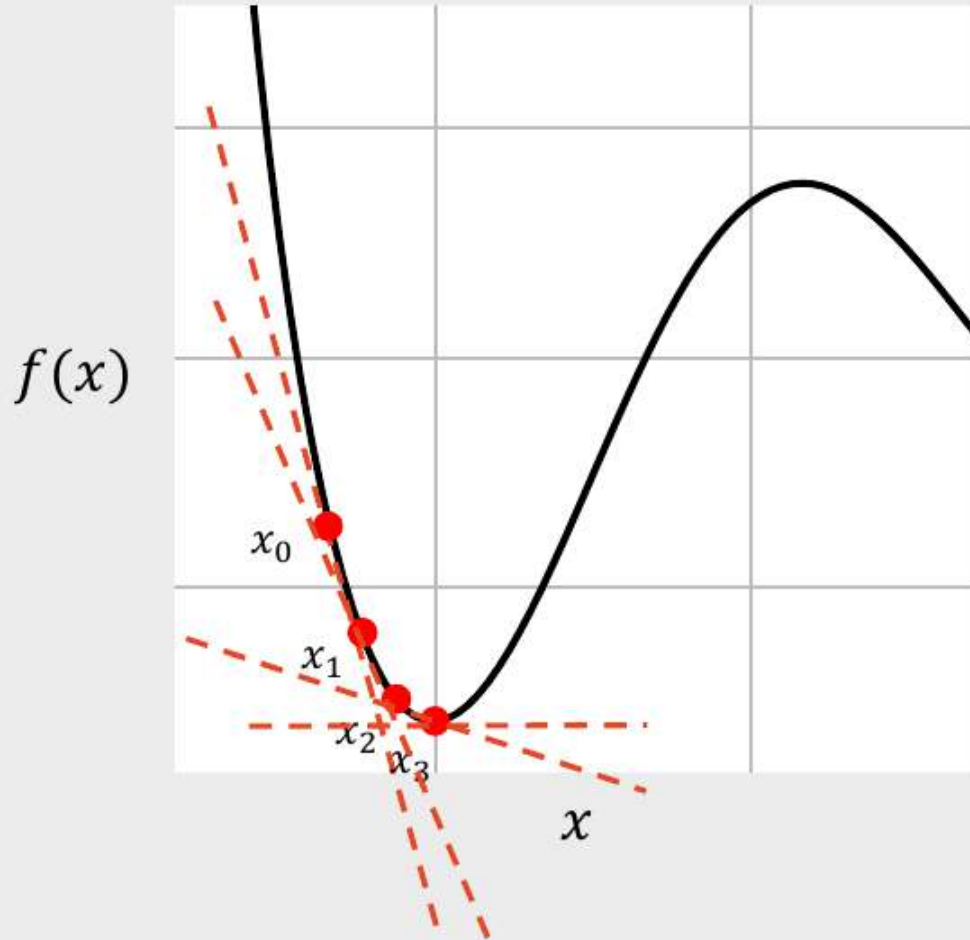
$$\frac{d^2f(x^*)}{dx^2} > 0$$

Global maximum Local maximum



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	<p>“Gradient”</p> $\nabla f(x_1, x_2, \dots, x_n)$ $= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]$ $= [0, 0, \dots, 0]$	<p>“Hessian”</p> $\nabla^2 f(x_1, x_2, \dots, x_n)$ $= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$ <p>Must be “positive definite”</p>	



Gradient Descent Method:

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 :

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

*Stop when $\nabla f(x_n) < \delta$ ↖ Very small number
or

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

Practice Question 2

Consider the following function:

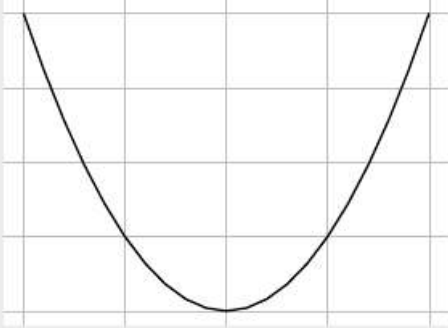
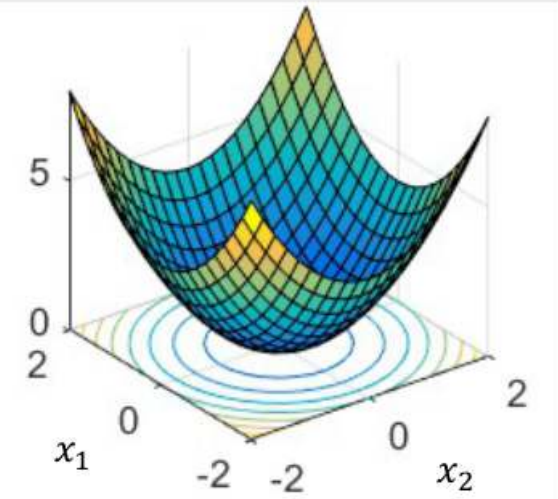
$$f(x) = x^2 - 6x$$

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	<p>“Gradient”</p> $\nabla f(x_1, x_2, \dots, x_n)$ $= \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]$ $= [0, 0, \dots, 0]$	<p>“Hessian”</p> $\nabla^2 f(x_1, x_2, \dots, x_n)$ $= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \dots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$ <p>Must be “positive definite”</p>	

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

emse-madd-gwu / `logitr-cars` Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

Clone ?

HTTPS SSH GitHub CLI

`https://github.com/emse-madd-gwu/logitr`

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

jhelvy finished mxl models		
code	finished mxl models	
data	finished mxl models	
figs	updated code to use {maddTools}	
models	finished mxl models	
sims	updated code to use {maddTools}	
.gitignore	added simulated data	3 months ago
README.md	Update README.md	last month
logitr-cars.Rproj	update Rproj file name	last month

Estimating utility models

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

Maximum likelihood estimation

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

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

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

$y_j = 1$ 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

Likely culprit: Merging two columns

```
head(names)
```

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

```
head(abbs)
```

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

```
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`

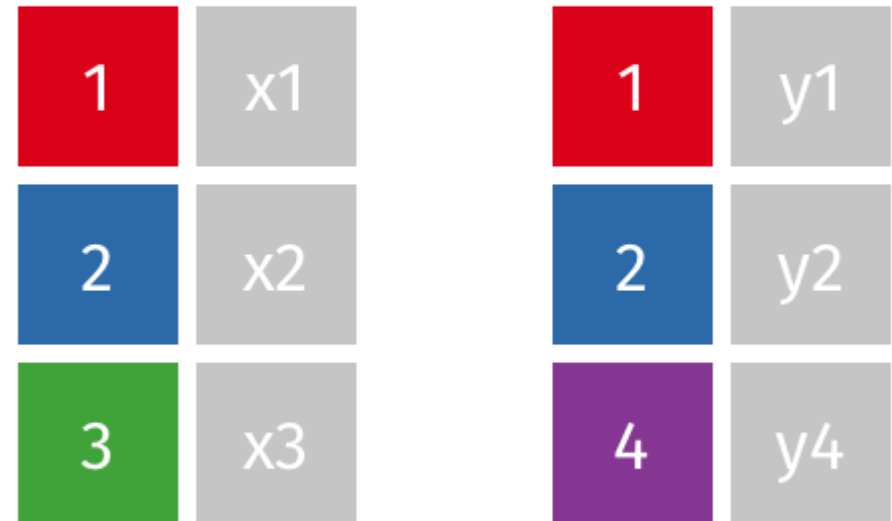
```
#> # A tibble: 3 × 2
#>   name plays
#>   <chr> <chr>
#> 1 John  guitar
#> 2 Paul  bass
#> 3 Keith guitar
```

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
```

inner_join(x, y)

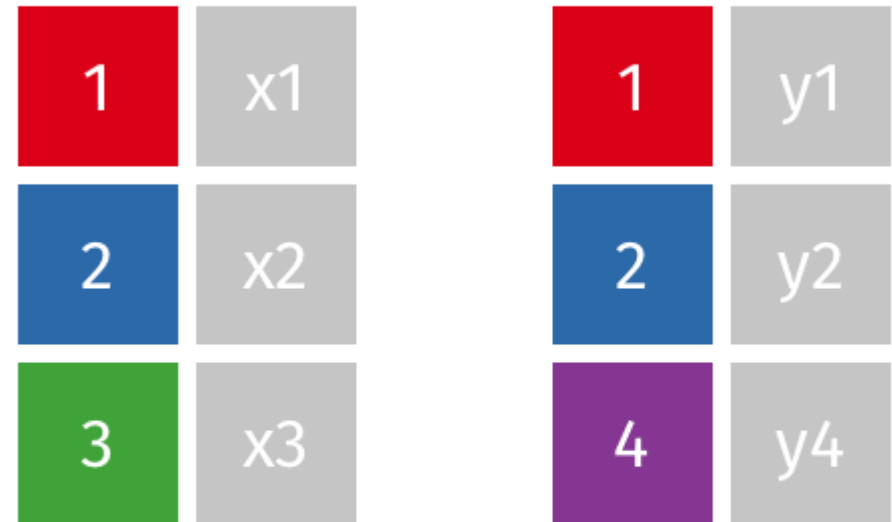


full_join()

```
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
```

full_join(x, y)

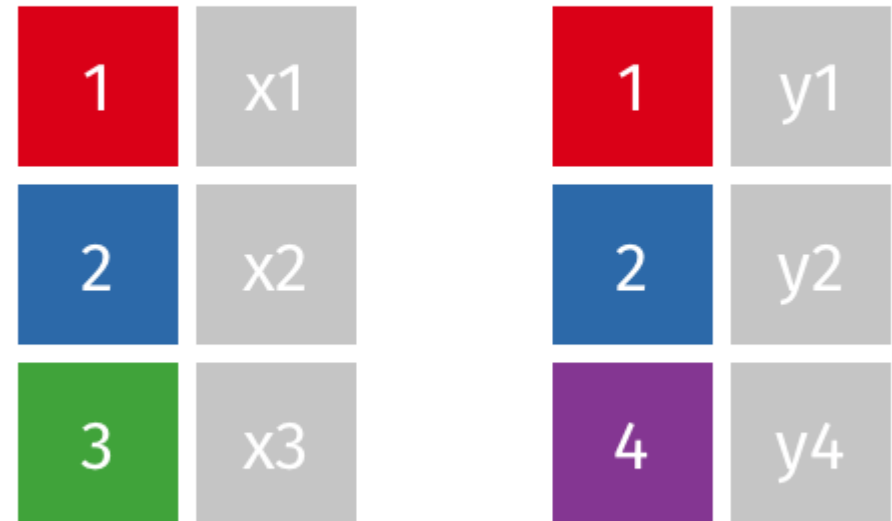


left_join()

```
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
```

left_join(x, y)

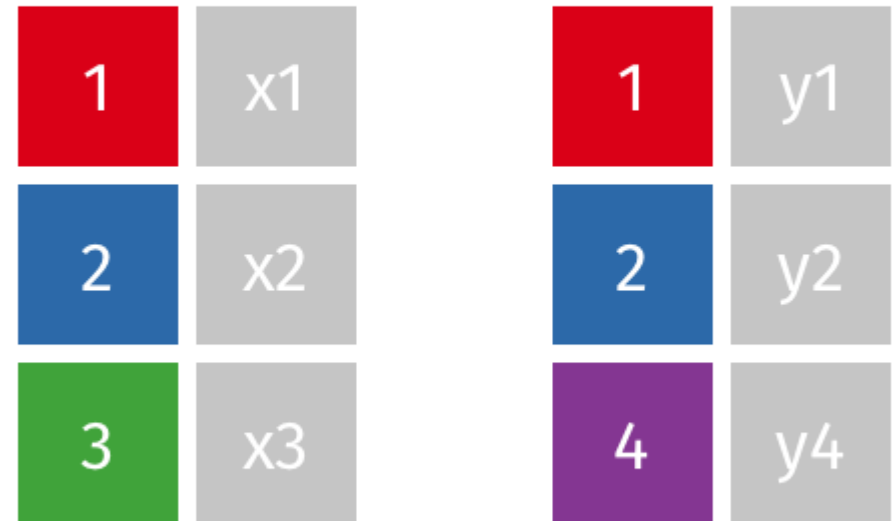


right_join()

```
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
```

right_join(x, y)



Specify the joining variable name

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

```
#> Joining with `by = join_by(name)`
```

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

```
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

1) Create a new data frame called `state_data` by joining the `state_abbs` and `state_regions` data frames. The result should be a data frame with variables `state_name`, `state_abb`, and `state_region`. It should look like this:

```
head(state_data)
```

```
#> # A tibble: 6 × 3
#>   state_name state_abb state_region
#>   <chr>      <chr>      <chr>
#> 1 Alabama    AL          Southeast
#> 2 Alaska     AK          Pacific
#> 3 Arizona    AZ          Mountain
#> 4 Arkansas  AR          Delta States
#> 5 California CA          Pacific
#> 6 Colorado  CO          Mountain
```

2) Join the `state_data` data frame to the `wildlife_impacts` data frame, adding the variables `state_region` and `state_name`.

```
glimpse(wildlife_impacts)
```

```
#> Rows: 56,978
#> Columns: 23
#> $ state_abb      <chr> "FL", "IN", NA, NA, NA, "FL", "FL", NA, NA, "FL",
#> $ state_name     <chr> "Florida", "Indiana", NA, NA, NA, "Florida", "Flo
#> $ state_region   <chr> "Southeast", "Corn Belt", NA, NA, NA, "Southeast"
#> $ incident_date  <dtm> 2018-12-31, 2018-12-29, 2018-12-29, 2018-12-27,
#> $ airport_id     <chr> "KMIA", "KIND", "ZZZZ", "ZZZZ", "ZZZZ", "KMIA", "
#> $ airport        <chr> "MIAMI INTL", "INDIANAPOLIS INTL ARPT", "UNKNOW
#> $ operator       <chr> "AMERICAN AIRLINES", "AMERICAN AIRLINES", "AMERIC
#> $ atype          <chr> "B-737-800", "B-737-800", "UNKNOWN", "B-737-900",
#> $ type_eng       <chr> "D", "D", NA, "D", "D", "D", "D", "D", "D", "D",
#> $ species_id     <chr> "UNKBL", "R", "R2004", "N5205", "J2139", "UNKB",
#> $ species        <chr> "Unknown bird - large", "Owls", "Short-eared owl",
#> $ damage         <chr> "M?", "N", NA, "M?", "M?", "N", "N", "N", "N", "N"
#> $ num_engs       <dbl> 2, 2, NA, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
#> $ incident_month <dbl> 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12,
#> $ incident_year  <dbl> 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018, 2018,
#> $ time_of_day    <chr> "Day", "Night", NA, NA, NA, "Day", "Night", NA, NA
#> $ time           <dbl> 1207, 2355, NA, NA, NA, 955, 948, NA, NA, 1321, 1
#> $ height         <dbl> 700, 0, NA, NA, NA, NA, 600, NA, NA, 0, NA, 0, NA
#> $ speed          <dbl> 200, NA, NA, NA, NA, NA, 145, NA, NA, 130, NA, NA
#> $ phase_of_flt  <chr> "Climb", "Landing Roll", NA, NA, NA, "Approach",
#> $ sky            <chr> "Some Cloud", NA, NA, NA, NA, "Some Cloud", NA
#> $ precip         <chr> "None", NA, NA, NA, NA, NA, "None", NA, NA, "None"
#> $ cost_repairs_infl_adj <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA
```

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

jhelvy / `formr4conjoint` Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

master 1 branch 0 tags Go to file Add file Code

jhelvy added package installs to readme

figs	added package installs to readme
survey	added consent form content in p1
.gitignore	Update .gitignore
LICENSE.md	Create LICENSE.md
README.Rmd	added package installs to readme
README.md	added package installs to readme 20 minutes ago
formr4conjoint.Rproj	Init 2 years ago

Clone ?
HTTPS SSH GitHub CLI
`https://github.com/jhelvy/formr4conjoi`

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

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