Chap 15: Missing Data and Other Opportunities

```
md"# Chap 15: Missing Data and Other Opportunities"
```

```
versioninfo()
```

```
Julia Version 1.11.0
Commit 501a4f25c2b (2024-10-07 11:40 UTC)
Build Info:
  Official https://julialang.org/ release
Platform Info:
  OS: Linux (x86_64-linux-gnu)
  CPU: 32 × Intel(R) Xeon(R) CPU E5-2630 v3 @ 2.40GHz
  WORD_SIZE: 64
  LLVM: libLLVM-16.0.6 (ORCJIT, haswell)
Threads: 16 default, 0 interactive, 8 GC (on 32 virtual cor
es)
Environment:
  JULIA_PKG_SERVER = https://mirrors.tuna.tsinghua.edu.cn/j
ulia
  JULIA_REVISE_WORKER_ONLY = 1
```

```
html"""
<style>
    main {
        margin: 0 auto;
        max-width: max(1800px, 75%);
        padding-left: max(5px, 1%);
        padding-right: max(350px, 10%);
    }
</style>
"""
```

Table of Contents

Chap 15: Missing Data and Other Opportunities

Code 15.1

15.1 Measurement error

Code 15.2

Code 15.3 model m15_1

Code 15.4

Code 15.5 model m15_2

Figure 15.2

Code 15.6 Figure 15.3

Code 15.7

15.2 Missing data

Code 15.8

```
begin
  using Pkg, DrWatson
  using PlutoUI
  TableOfContents()
end
```

```
begin
    using Turing
    using Turing
    using DataFrames
    using CSV
    using Random
    using Dagitty
    using Distributions
    #using StatisticalRethinking
    using StatisticalRethinking: link
    using StatisticalRethinkingPlots
    using StatsPlots
    using StatsBase
    using Logging
    using LinearAlgebra
end
```

Code 15.1

```
md"## Code 15.1"
```

```
0.6617857711284418
```

```
begin
    Random.seed!(2)

    function sim_pancake()
        pancake = [[1, 1], [1, 0], [0, 0]]
        sides = sample(pancake)
        sample([sides, reverse(sides)])
    end

    @time pancakes = vcat([sim_pancake() for _ in 1:100_000]'...)
    up = pancakes[:,1]
    down = pancakes[:,2]

    num_11_10 = sum(up .== 1)
    num_11 = sum((up .== 1) .& (down .== 1))
    num_11 / num_11_10
end
```

```
  0.103752 seconds (1.65 M allocations: 64.906 MiB, 58.
63% compilation time)
```

```
pancake =  [[1, 1], [1, 0], [0, 0]]
```

```
pancake = [[1, 1], [1, 0], [0, 0]]
```

```
sides =  [0, 0]
```

```
sides = sample(pancake)
```

```
[0, 0]
```

```
sample([sides, reverse(sides)])
```

```
[[0, 0], [0, 0]]
```

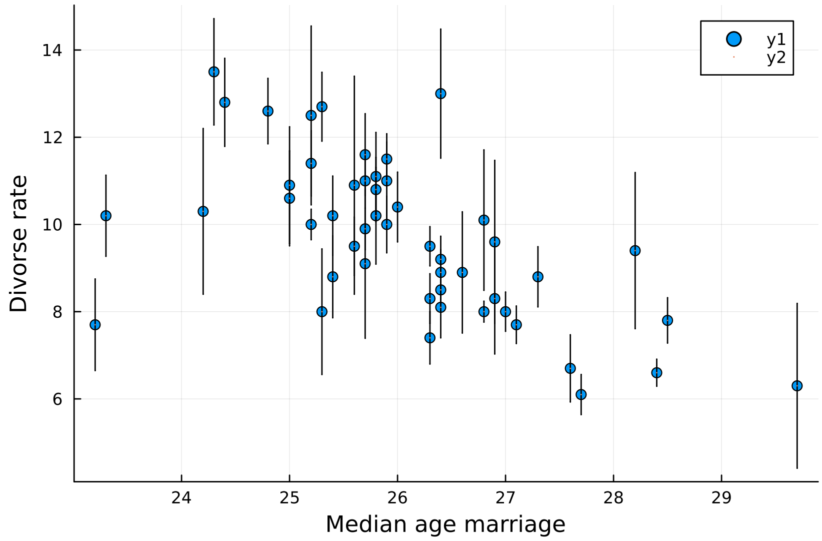
```
[sides, reverse(sides)]
```

15.1 Measurement error

```
md" # 15.1 Measurement error"
```

Code 15.2

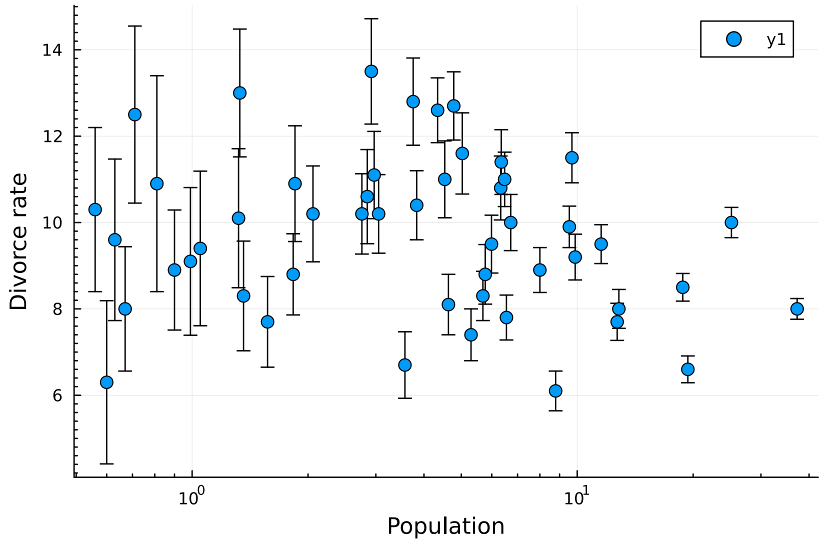
```
md"## Code 15.2"
```



```
begin
    d_divorce = DataFrame(CSV.File("data/WaffleDivorce.csv"))

    scatter(d_divorce.MedianAgeMarriage, d_divorce.Divorce,
        xlab="Median age marriage", ylab="Divorse rate")
    scatter!(d_divorce.MedianAgeMarriage, d_divorce.Divorce,
    yerror=d_divorce."Divorce SE", ms=0)
end
```

	Location	Loc	Population	MedianAgeMarriage	Marriage	Ma
1	"Alabama"	"AL"	4.78	25.3	20.2	1.
2	"Alaska"	"AK"	0.71	25.2	26.0	2.
3	"Arizona"	"AZ"	6.33	25.8	20.3	0.

```
first(d_divorce,3)
```



```
begin
    scatter(d_divorce.Population, d_divorce.Divorce, xaxis=:log10,
        xlab="Population", ylab="Divorce rate",
        xminorticks=9, yminorticks=10,
        yerror=d_divorce."Divorce SE", ms=5)
    #scatter!(d_divorce.Population, d_divorce.Divorce, yerror=d_divorce."Divorce SE", ms=0)
end
```

Code 15.3 model `m15_1`

```
md"## Code 15.3 model `m15_1`"
```

	D_true[10]	D_true[11]	D_true[12]	D_true[13]	D_true[14]	D_t
1	-0.784182	0.591328	-0.412152	0.12795	-0.820124	0.9
2	-0.418201	1.43575	-0.850778	0.108591	-0.808901	0.5
3	-0.638492	0.392377	-0.361222	1.00863	-0.986006	0.6
4	-0.509696	1.10383	-0.518881	0.27218	-0.814202	0.5
5	-0.691987	0.769628	-0.666241	0.826797	-0.71571	0.4
6	-0.521323	0.589482	-0.991947	-0.00510027	-1.00075	0.6
7	-0.586819	0.689822	-0.252866	0.751031	-0.657953	0.4
8	-0.582998	0.455282	-0.310074	0.56882	-0.711395	0.5
9	-0.630576	0.790008	-0.888668	1.30604	-0.891928	0.4
10	-0.630576	0.790008	-0.888668	1.30604	-0.891928	0.4
more						
1000	-1.02493	0.883494	-0.144567	-0.00200293	-0.840648	0.6

```julia
begin
    d_divorce_ls = (
        D_obs = standardize(ZScoreTransform, d_divorce.Divorce),
        D_sd = d_divorce."Divorce SE" ./ std(d_divorce.Divorce),
        M = standardize(ZScoreTransform, d_divorce.Marriage),
        A = standardize(ZScoreTransform, d_divorce.MedianAgeMarriage),
        N = nrow(d_divorce),
    )

    @model function m15_1(D_obs, D_sd, M, A, N)
        a ~ Normal(0, 0.2)
        bA ~ Normal(0, 0.5)
        bM ~ Normal(0, 0.5)
        μ = @. a + bA * A + bM * M
        σ ~ Exponential()
        D_true ~ MvNormal(μ, σ)
        @. D_obs ~ Normal(D_true, D_sd)
    end

    Random.seed!(1)
    @time m15_1_ch = sample(m15_1(d_divorce_ls...), NUTS(), 1000)
    m15_1_df = DataFrame(m15_1_ch);
end
```

```
Sampling 100%

Found initial step size
ϵ: 0.2
```

```
11.256042 seconds (16.66 M allocations: 6.157 GiB, 10.68% gc time, 55.28% compilation time)
```

Code 15.4

```julia
md"## Code 15.4"
```

	variable	mean	min	median	max
1	Symbol("D_true[10]")	-0.622426	-1.17513	-0.621466	-0.05984
2	Symbol("D_true[11]")	0.752743	-0.167793	0.764524	1.76655
3	Symbol("D_true[12]")	-0.54162	-2.09472	-0.538969	1.42518
4	Symbol("D_true[13]")	0.191023	-1.80048	0.197183	1.54803
5	Symbol("D_true[14]")	-0.86873	-1.59464	-0.878422	-0.13569
6	Symbol("D_true[15]")	0.563774	-0.450136	0.559766	1.55619
7	Symbol("D_true[16]")	0.269308	-0.855484	0.282876	1.57184
8	Symbol("D_true[17]")	0.505615	-0.78145	0.504514	1.83022
9	Symbol("D_true[18]")	1.25328	0.14058	1.25724	2.48261
10	Symbol("D_true[19]")	0.428978	-0.812482	0.441281	1.63373
	more				
54	:σ	0.579131	0.30084	0.575787	1.00322

```julia
describe(m15_1_df)
```

Code 15.5 model `m15_2`

```julia
md"## Code 15.5 model `m15_2`"
```

```
(D_obs = [1.65421, 1.54436, 0.610716, 2.09357, -0.927058, 1.05008, -1.
```

```julia
begin
    dlist2 = (
        D_obs = standardize(ZScoreTransform, d_divorce.Divorce),
        D_sd = d_divorce."Divorce SE" ./ std(d_divorce.Divorce),
        M_obs = standardize(ZScoreTransform, d_divorce.Marriage),
        M_sd = d_divorce."Marriage SE" ./ std(d_divorce.Marriage),
        A = standardize(ZScoreTransform, d_divorce.MedianAgeMarriage),
        N = nrow(d_divorce),
    )
end
```

```
[0.083057, 1.01903, 0.0594721, 1.41732, -0.266635, 0.830463, -0.76543
```

```
begin

    @model function m15_2(D_obs, D_sd, M_obs, M_sd, A, N)
        a ~ Normal(0, 0.2)
        bA ~ Normal(0, 0.5)
        bM ~ Normal(0, 0.5)
        M_true ~ filldist(Normal(), N)

        μ = @. a + bA * A + bM * M_true
        σ ~ Exponential()
        D_true ~ MvNormal(μ, σ)
        @. D_obs ~ Normal(D_true, D_sd)
        @. M_obs ~ Normal(M_true, M_sd)
    end

    Random.seed!(1)
    @time m15_2_ch = sample(m15_2(dlist2...), NUTS(), 1000)
    m15_2_df = DataFrame(m15_2_ch);
    D_true = [mean(m15_2_df[!, "D_true[$i]"]) for i ∈
    1:dlist2.N]
    M_true = [mean(m15_2_df[!, "M_true[$i]"]) for i ∈
    1:dlist2.N]
end
```

```
 46.280141 seconds (67.04 M allocations: 42.705 GiB, 1
7.65% gc time, 36.69% compilation time)
 35.882352 seconds (53.92 M allocations: 42.077 GiB, 19.90%
gc time, 21.32% compilation time)
```

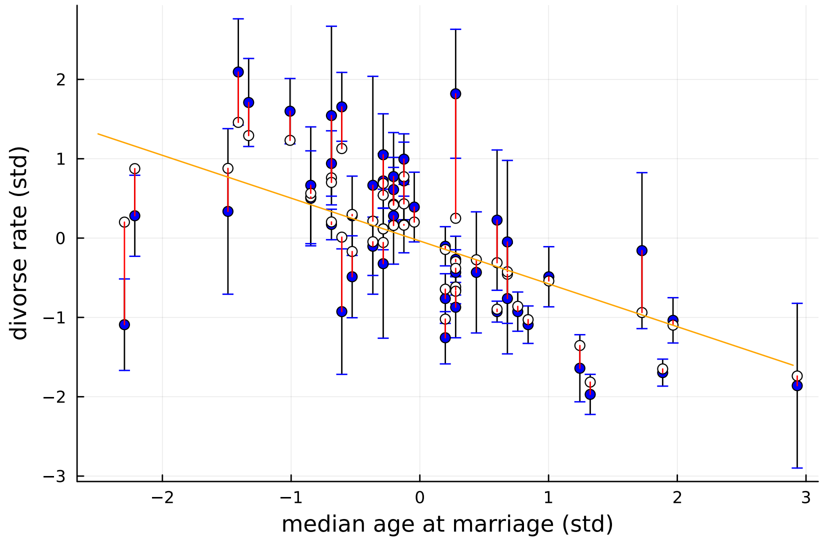
```
Sampling 100%

Found initial step size
ϵ: 0.4
```

	variable	mean	min	median	max
1	Symbol("D_true[10]")	-0.616598	-1.09836	-0.616169	-0.10918
2	Symbol("D_true[11]")	0.773391	-0.153289	0.772106	1.59042
3	Symbol("D_true[12]")	-0.455932	-1.96422	-0.469349	1.27627
4	Symbol("D_true[13]")	0.201203	-1.44406	0.204312	1.67876
5	Symbol("D_true[14]")	-0.860255	-1.57298	-0.85922	-0.15458
6	Symbol("D_true[15]")	0.540992	-0.540644	0.543722	1.62189
7	Symbol("D_true[16]")	0.297736	-0.943139	0.293591	1.44996
8	Symbol("D_true[17]")	0.519618	-1.31079	0.522772	2.32168
9	Symbol("D_true[18]")	1.23177	0.22005	1.22341	2.29087
10	Symbol("D_true[19]")	0.431547	-0.906202	0.416142	1.98877
	more				
104	:σ	0.563163	0.242072	0.558362	0.974335

```
describe(m15_2_df)
```

Figure 15.2

```
md"## Figure 15.2"
```

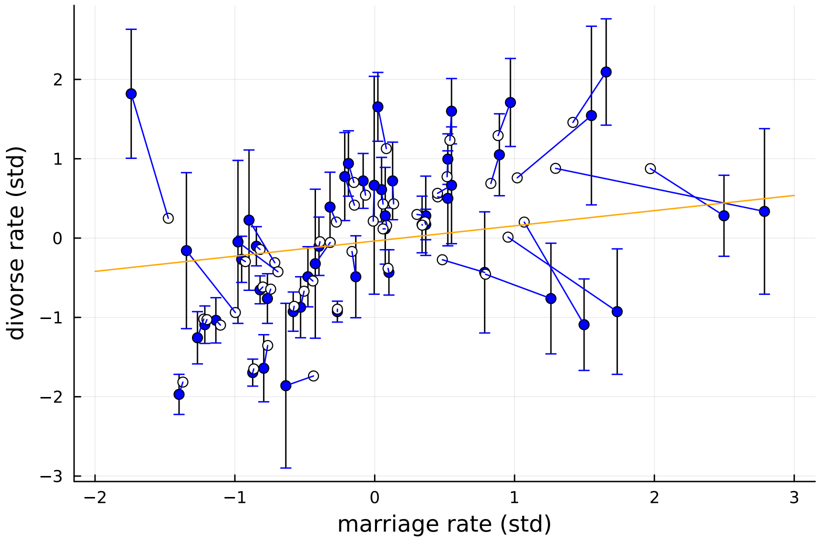


```julia
begin

    p1 = scatter(dlist2.A, dlist2.D_obs, mc=:blue, yerror=dlist2.D_sd,
        label="observed", xlab="median age at marriage (std)", ylab="divorse rate (std)")
    scatter!(dlist2.A, D_true, mc=:white, label="true")

    for i ∈ 1:dlist2.N
        plot!([dlist2.A[i], dlist2.A[i]], [dlist2.D_obs[i], D_true[i]], c=:red, legend=false)
    end
    x = -2.5:0.2:3
    y = -0.0368595 .+  -0.540089 .* x
    plot!(x,y, c=:orange, label="m15_2 estimate")
    p1
end
```

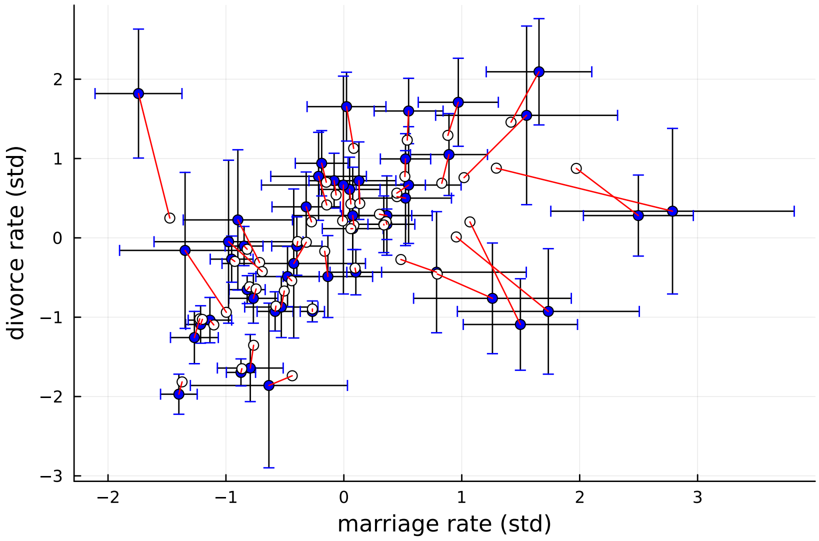
```
Enter cell code...
```

Code 15.6 Figure 15.3

```julia
md"## Code 15.6 Figure 15.3"
```



```julia
begin
    p2 = scatter(dlist2.M_obs, dlist2.D_obs, mc=:blue, yerror=dlist2.D_sd,
        label="observed", xlab="marriage rate (std)", ylab="divorse rate (std)",
        legend=true)
    scatter!(M_true, D_true, mc=:white, label="true", legend=true)

    for i ∈ 1:dlist2.N
        plot!([dlist2.M_obs[i], M_true[i]], [dlist2.D_obs[i], D_true[i]], c=:blue, legend=false)
    end
    x2 = -2:0.2:3
    y2 = -0.0368595 .+  0.1915 .* x2
    plot!(x2,y2, c=:orange, label="m15_2 estimate")
    p2
end
```



```julia
begin
    p3 = scatter(dlist2.M_obs, dlist2.D_obs, mc=:blue, xerror=dlist2.M_sd, yerror=dlist2.D_sd,
        label="observed", xlab="marriage rate (std)", ylab="divorse rate (std)")
    scatter!(M_true, D_true, mc=:white, label="true")

    for i ∈ 1:dlist2.N
        plot!([dlist2.M_obs[i], M_true[i]], [dlist2.D_obs[i], D_true[i]], c=:red, legend=false)
    end
    p3
end
```

```
Enter cell code...
```

Code 15.7

```
md"## Code 15.7"
```

```
[-0.366839, -2.48606, 0.579584, -0.588886, -1.54843, -2.13782, -1.197
```

```
let
    N = 500
    A = rand(Normal(), N)
    M = rand.(Normal.(-A))
    D = rand.(Normal.(A))
    A_obs = rand.(Normal.(A));
end
```

15.2 Missing data

```
md"# 15.2 Missing data"
```

```
Enter cell code...
```

Code 15.8

```
md"## Code 15.8"
```

```
[4, 8, 0, 7, 6, 5, 4, 5, 5, 6, 8, 7, 6, 9, 4, 4, 8, 8, 8, 1,    more ,4, 6
```

```
let
    N = 100
    S = rand(Normal(), N)
    H = rand.([BinomialLogit(10, l) for l in S]);
end
```