Plotting data in DataFrames

INTRODUCTION TO DATA VISUALIZATION WITH JULIA

Gustavo Vieira Suñe
Data Analyst

datacamp

Insurance dataset

- `insurance` DataFrame

Age	Sex	BMI	Children	Smoker	Region	Charges
19	female	27.90	0	yes	southwest	16884.90
18	male	33.77	1	no	southeast	1725.55
28	male	33.00	3	no	southeast	4449.46
...	...	...	...	...	...	...

- DataFrames are flexible and efficient for tabular data
  - `StatsPlots` has a recipe to plot data in DataFrames
  - Introducing the `@df` notation!

Extracting arrays from DataFrame

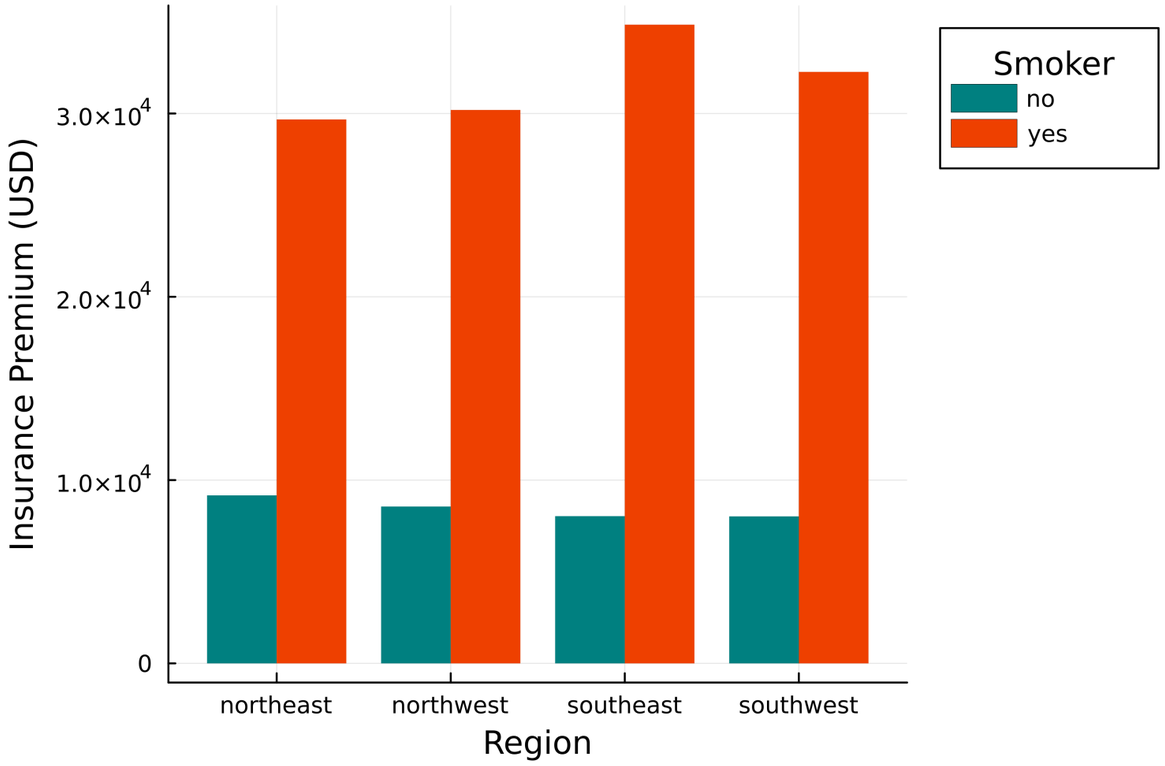
- Mean charges by region and smoker status

```
# Group by region and smoker
grouped = groupby(insurance, [:Region, :Smoker])
# Calculate mean charges
grouped_mean_charges = combine(grouped, :Charges => mean)
```

- Each column gives an array of data
  - For example, `grouped_mean_charges.Region` extracts an array containing the regions as strings.

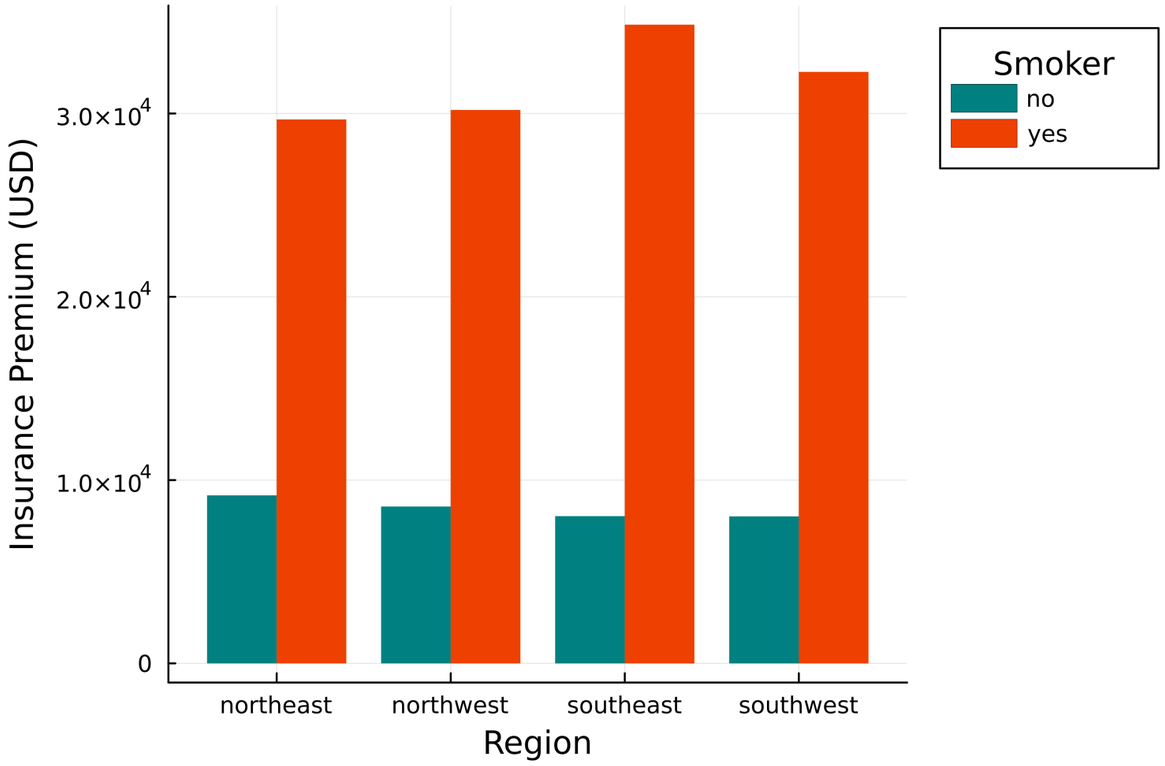
Plotting data in arrays

```
# Grouped bar chart
groupedbar(
    # Pass arrays as arguments
    grouped_mean_charges.Region,
    grouped_mean_charges.Charges_mean,
    group=grouped_mean_charges.Smoker,
    color=[:teal :orangered2],
    linewidth=0,
    legend_title="Smoker",
    legend_position=:outertopright)
xlabel!("Region")
ylabel!("Insurance Premium (USD)")
```



Plotting from DataFrames directly

```julia
# Plot from DataFrame
@df grouped_mean_charges groupedbar(
    # Pass column names
    :Region,
    :Charges_mean,
    group=:Smoker,
    color=[:teal :orangered2],
    linewidth=0,
    legend_title="Smoker",
    legend_position=:outertopright)
xlabel!("Region")
ylabel!("Insurance Premium (USD)")
```



Side-by-side comparison

```julia
# Grouped bar chart
groupedbar(
    # Pass arrays as arguments
    grouped_mean_charges.Region,
    grouped_mean_charges.Charges_mean,
    group=grouped_mean_charges.Smoker,
    color=[:teal :orangered2],
    linewidth=0,
    legend_title="Smoker",
    legend_position=:outertopright)
xlabel!("Region")
ylabel!("Insurance Premium (USD)")
```

```julia
# Plot from DataFrame
@df grouped_mean_charges groupedbar(
    # Pass column names
    :Region,
    :Charges_mean,
    group=:Smoker,
    color=[:teal :orangered2],
    linewidth=0,
    legend_title="Smoker",
    legend_position=:outertopright)
xlabel!("Region")
ylabel!("Insurance Premium (USD)")
```

Chaining DataFrame commands

- From before

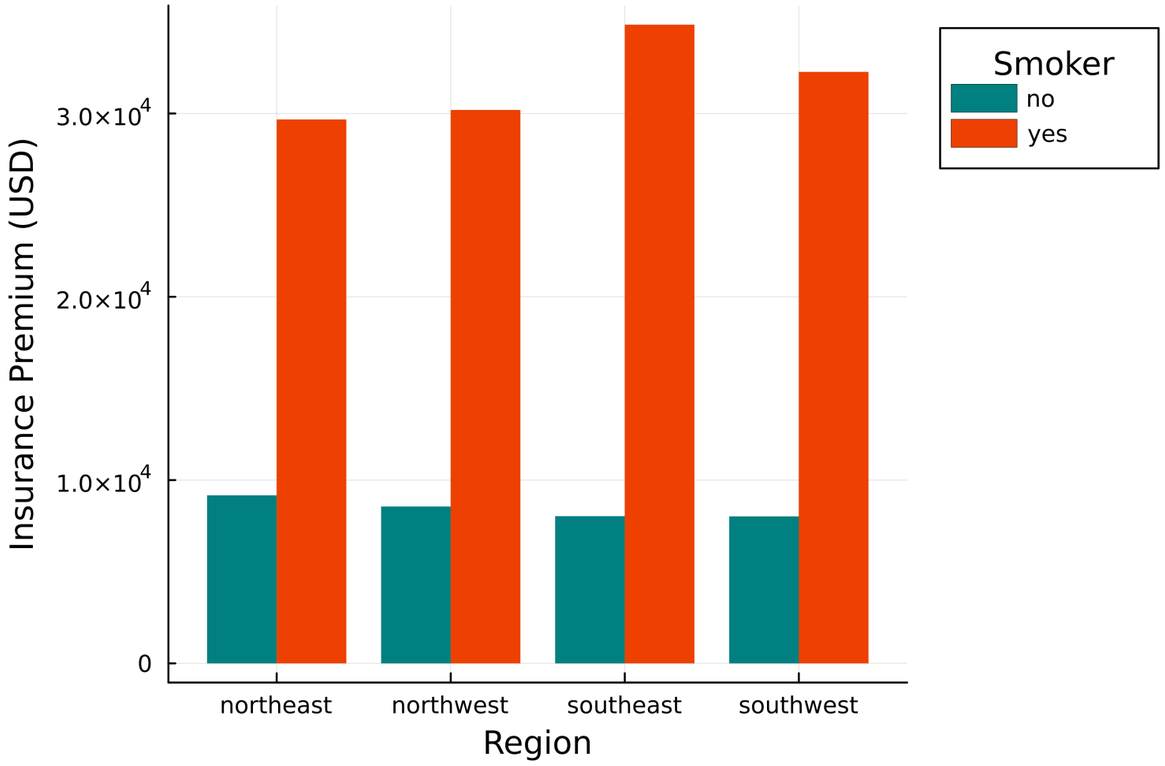
```
# Group by region and smoker
grouped = groupby(insurance, [:Region, :Smoker])
# Calculate mean charges
grouped_mean_charges = combine(grouped, :Charges => mean)
```

- Use chaining instead

```
using Chain
# Chain groupby and combine
grouped_mean_charges = @chain insurance begin
  groupby([:Region, :Smoker])
  combine(:Charges => mean)
end
```

Plotting chain

```
# Plotting chain
@chain insurance begin
    # Manipulate data
    groupby([:Region, :Smoker])
    combine(:Charges => mean)
    # Plot data
    @df groupedbar(:Region,
        :Charges_mean,
        group=:Smoker,
        color=[:teal :orangered2],
        linewidth=0, legend_title="Smoker",
        legend_position=:outertopright)
end
xlabel!("Region")
ylabel!("Insurance Premium (USD)")
```



Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH JULIA

Plotting in More Dimensions

INTRODUCTION TO DATA VISUALIZATION WITH JULIA

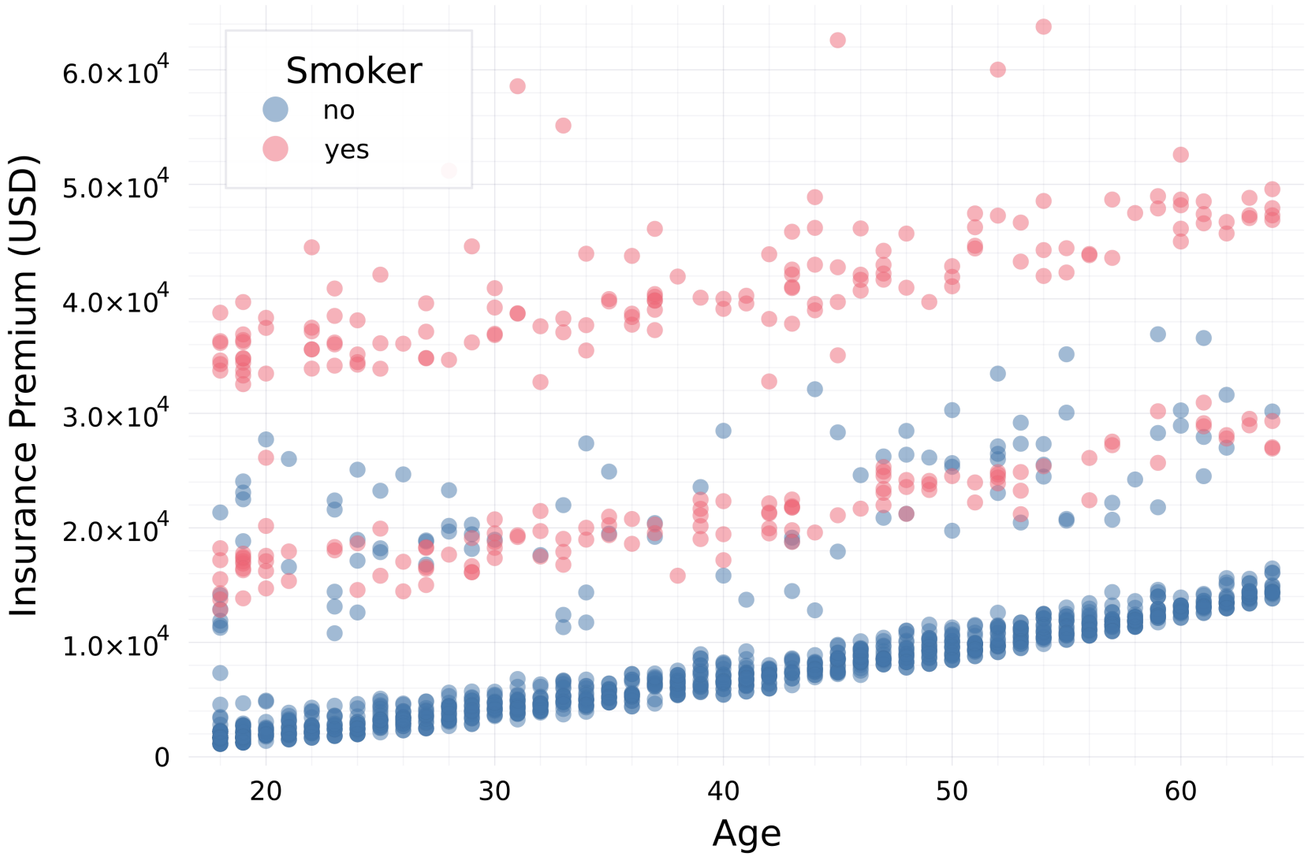
Gustavo Vieira Suñe
Data Analyst

datacamp

Why more dimensions?

- Recognize underlying patterns and trends
- Analyze relationships between multiple variables simultaneously
- Identify clusters
- Clear and immersive presentation
- Feature engineering

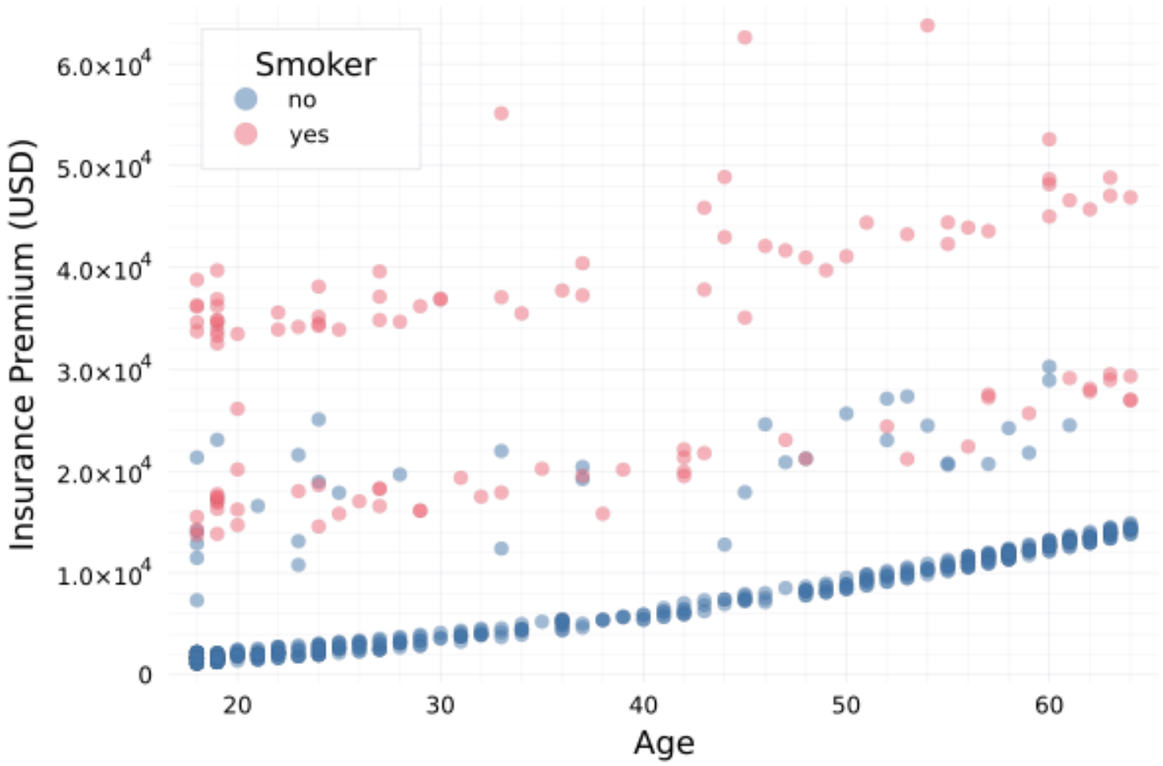
Will clusters persist?



- Are these clusters present for any number of children?

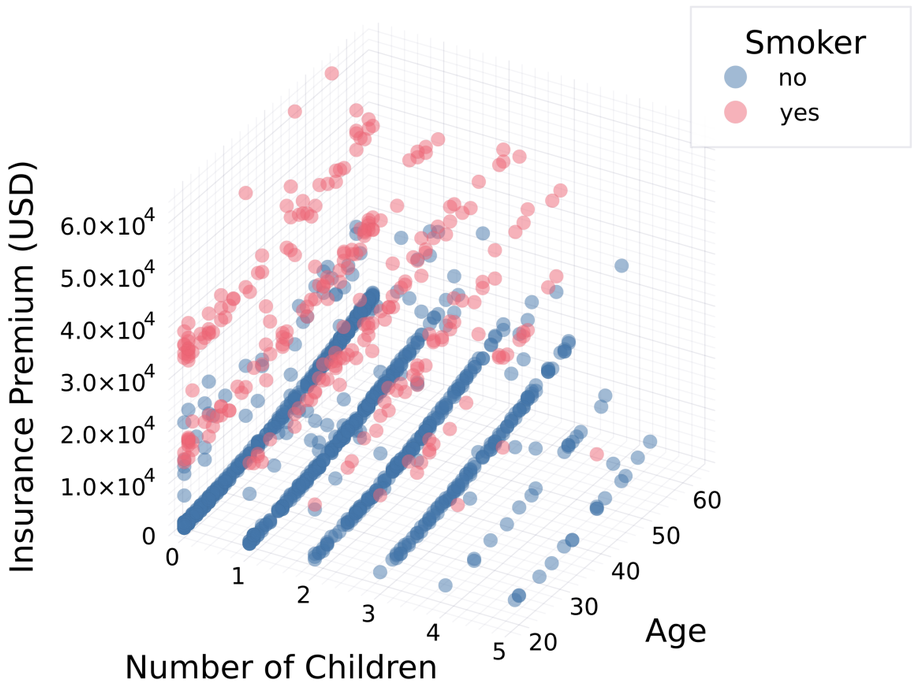
Plotting a slice

```
theme(:bright)

# Filter data
no_children = filter(
    row -> row.Children == 0, insurance)

# Plot a slice
@df no_children scatter(
    :Age, :Charges, group=:Smoker,
    markersize=4, alpha=0.5,
    legend_title="Smoker")
xlabel!("Age")
ylabel!("Insurance Premium (USD)")
```



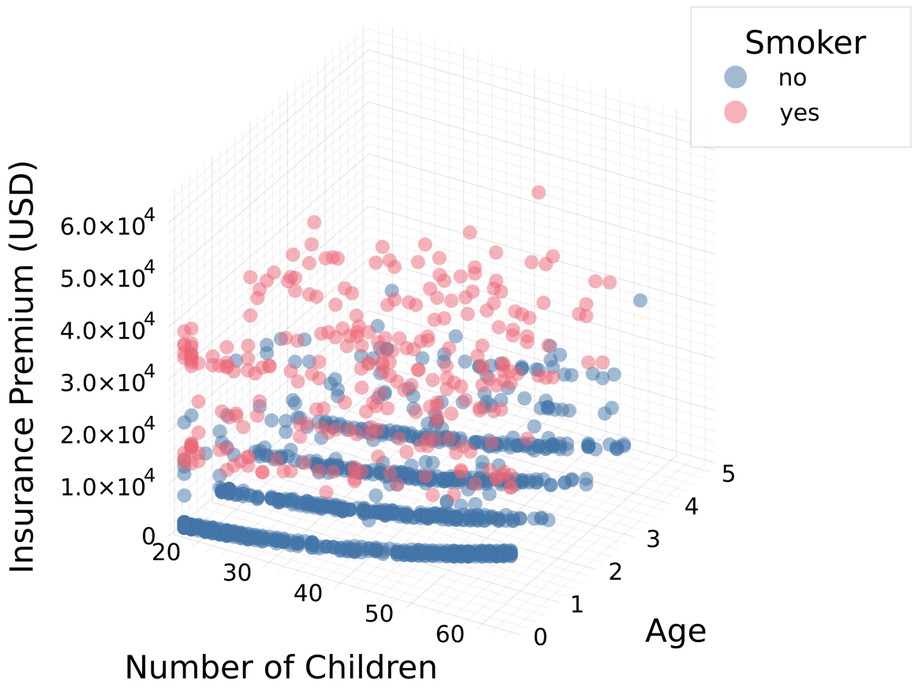
Using another dimension

```
theme(:bright)
@df insurance scatter(
    # Pass three columns
    :Children,
    :Age,
    :Charges,
    group=:Smoker,
    markersize=4, alpha=0.5,
    legend_title="Smoker"
)
# Axis labels
xlabel!("Number of Children")
ylabel!("Age")
zlabel!("Insurance Premium (USD)")
```

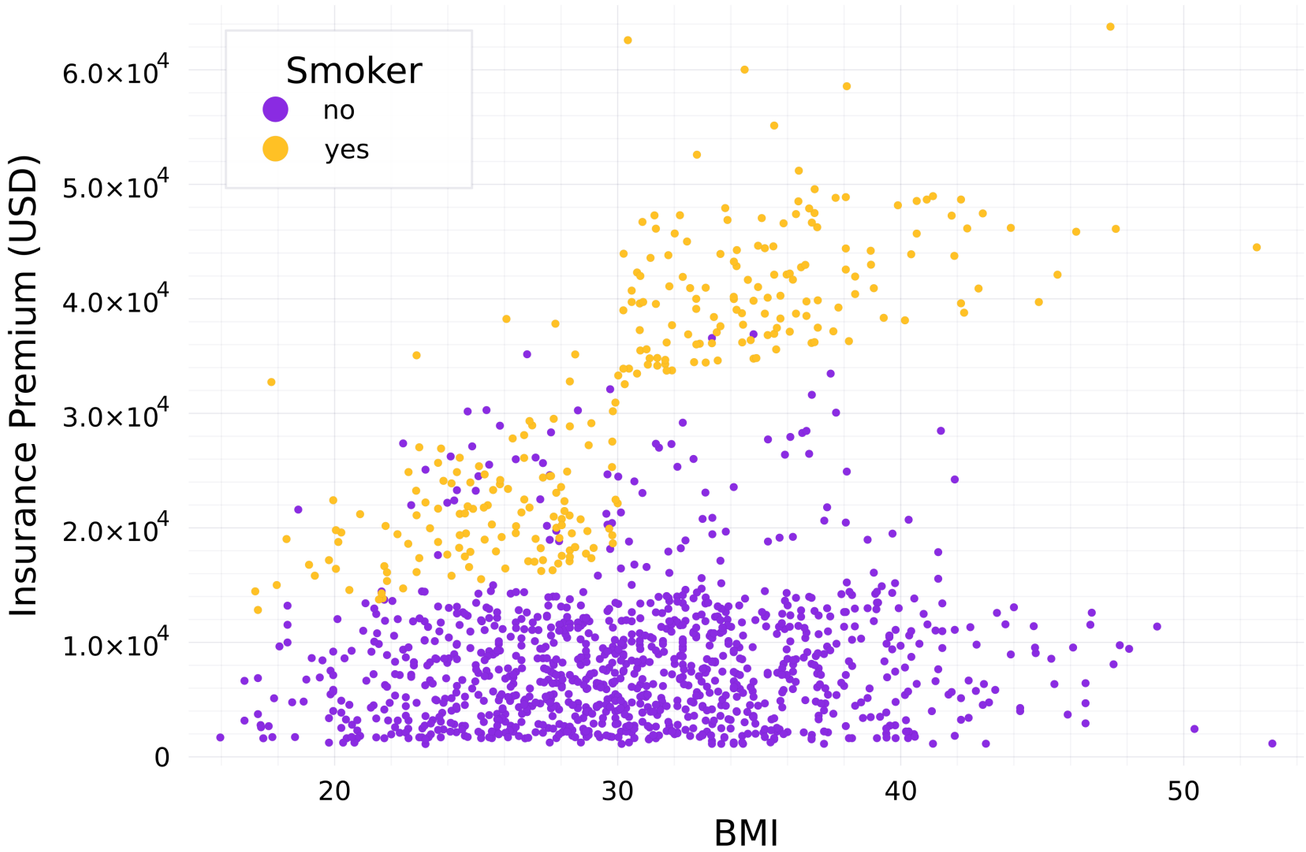


Axis order

```
theme(:bright)
@df insurance scatter(
    # Swap :Age and :Children
    :Age,
    :Children,
    :Charges,
    group=:Smoker,
    markersize=4, alpha=0.5,
    legend_title="Smoker"
)
# Axis labels
xlabel!("Number of Children")
ylabel!("Age")
```



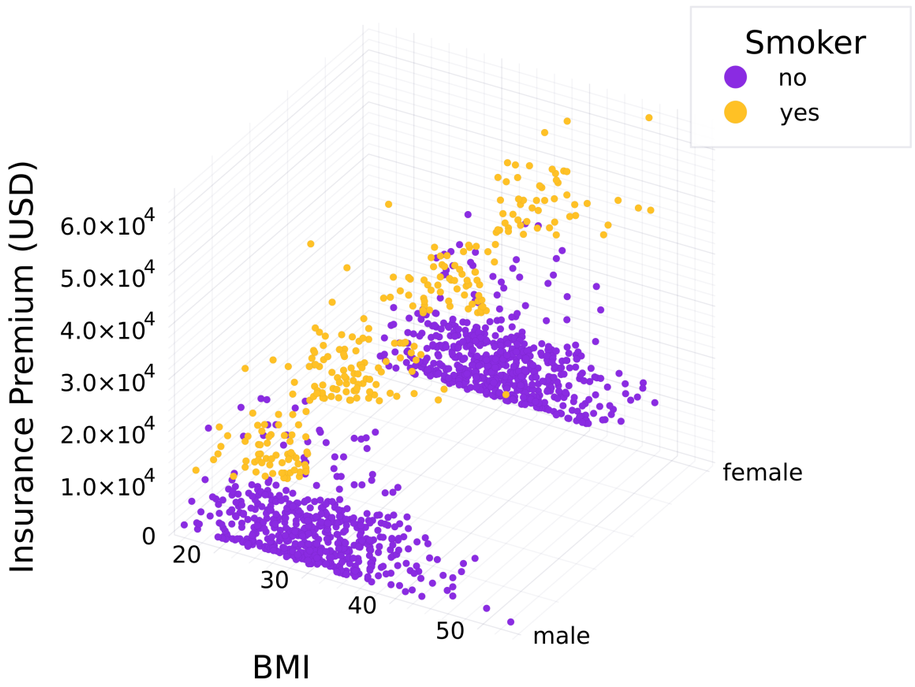
Grouping by another category



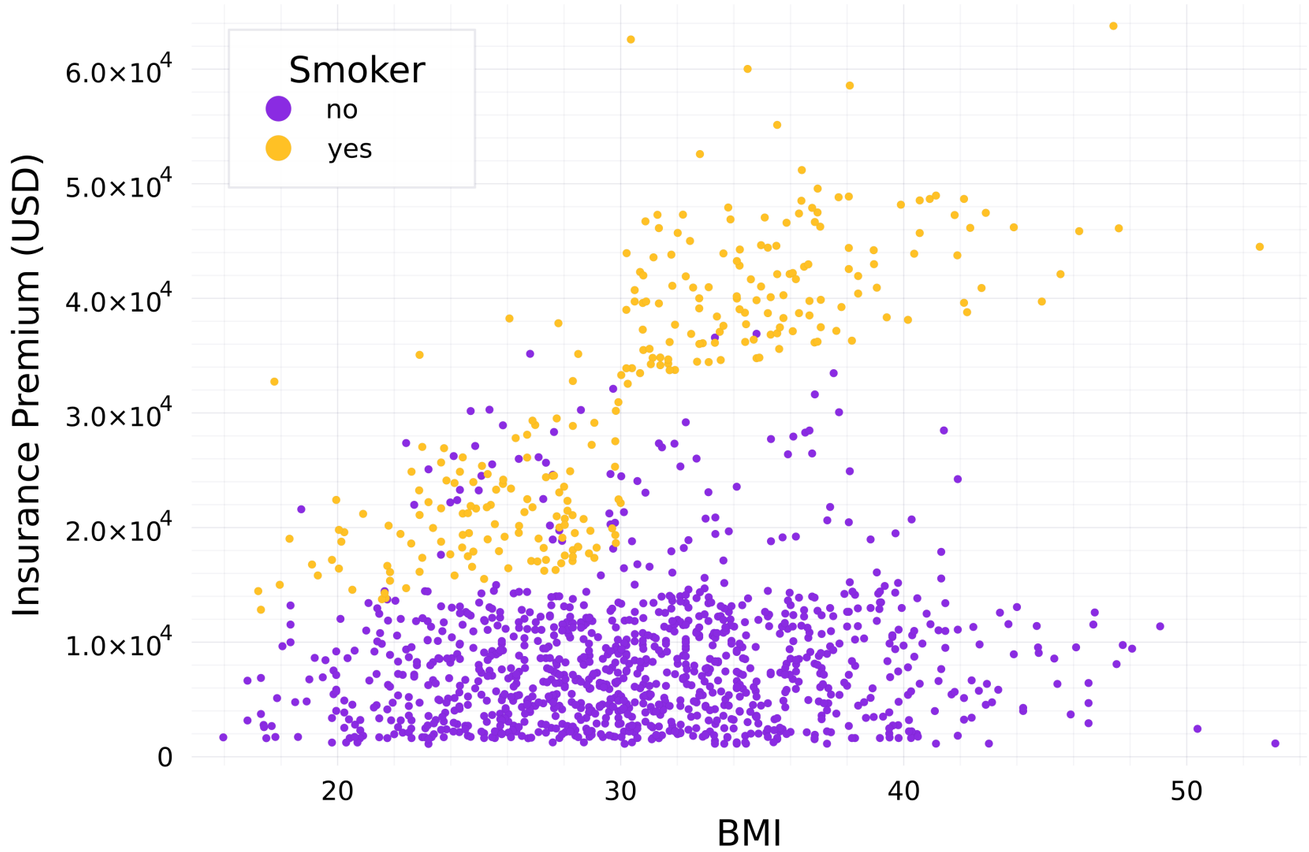
- Can we group by smoker status and sex?

Add a categorical dimension

```
theme(:vibrant)

@df insurance scatter(
    :BMI,
    # Pass categorical column
    :Sex,
    :Charges,
    group=:Smoker,
    markersize=2, legend_title="Smoker",
    color=[:blueviolet :goldenrod1])
xlabel!("BMI")
zlabel!("Insurance Premium (USD)")
```



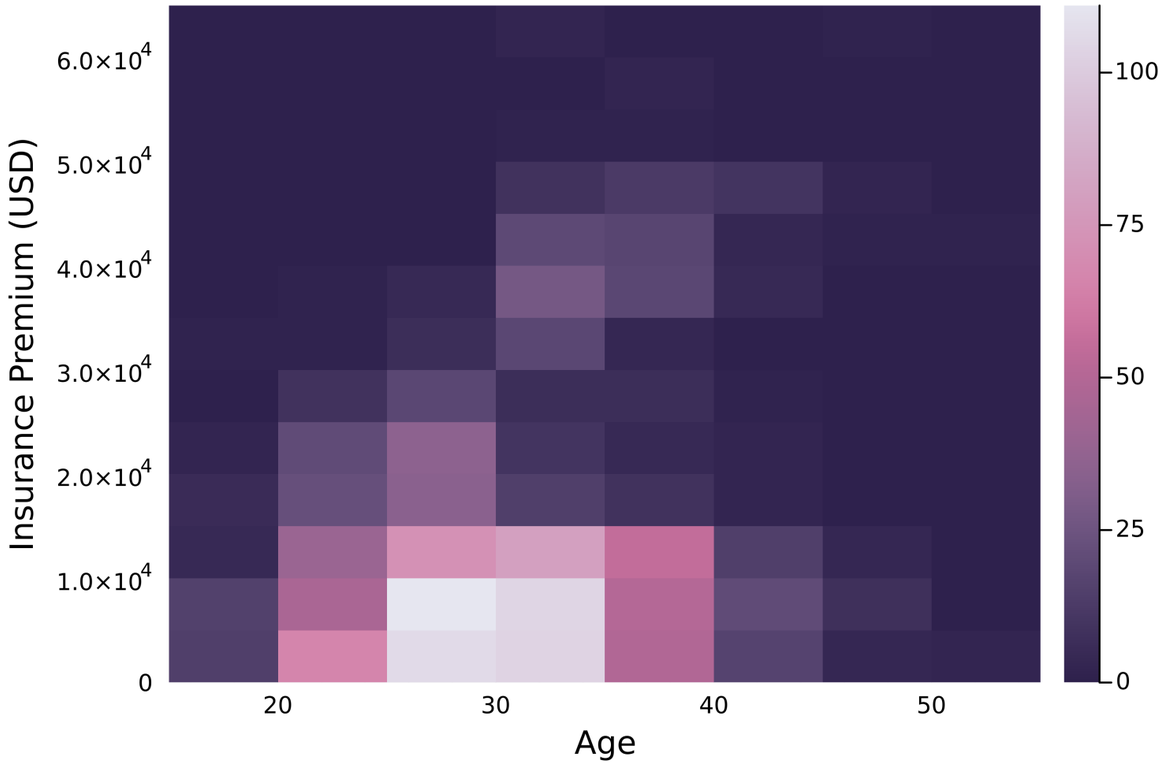
Visualize point density



- Can we visualize the point density more clearly?

Two-dimensional histograms

```
# 2d histogram
@df insurance histogram2d(
    :BMI,
    :Charges,
    # Fill color scheme
    fillcolor=:acton,
    # Fill empty bins
    show_empty_bins=true,
)
xlabel!("Age")
ylabel!("Insurance Premium (USD)")
```



Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH JULIA

Multiple plots from DataFrames

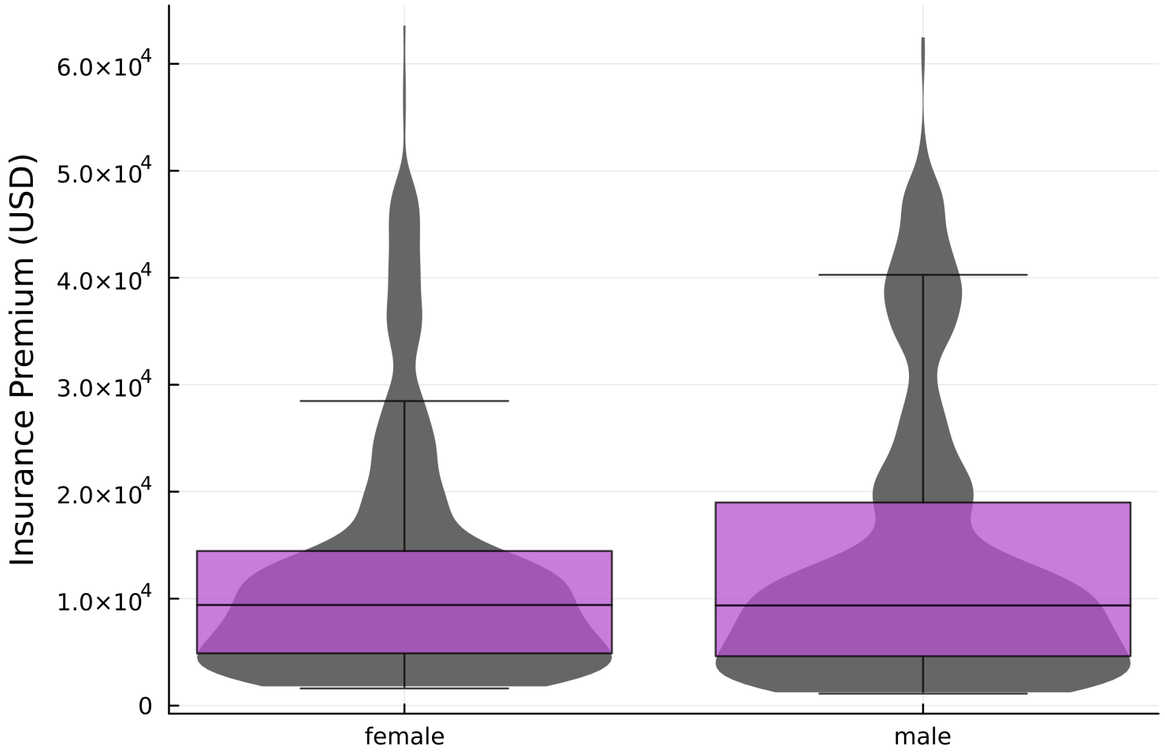
INTRODUCTION TO DATA VISUALIZATION WITH JULIA

Gustavo Vieira Suñe
Data Analyst

datacamp

Multiple variables in a plot

```
# Violin plot
@df insurance violin(
    :Sex, :Charges,
    label=false, linewidth=0,
    fillcolor=:grey40
)
# Add box plot
@df insurance boxplot!(
    :Sex, :Charges,
    label=false, alpha=0.75,
    fillcolor=:mediumorchid3,
    outliers=false,
)
ylabel!("Insurance Premium (USD)")
```



Categorical data and layouts

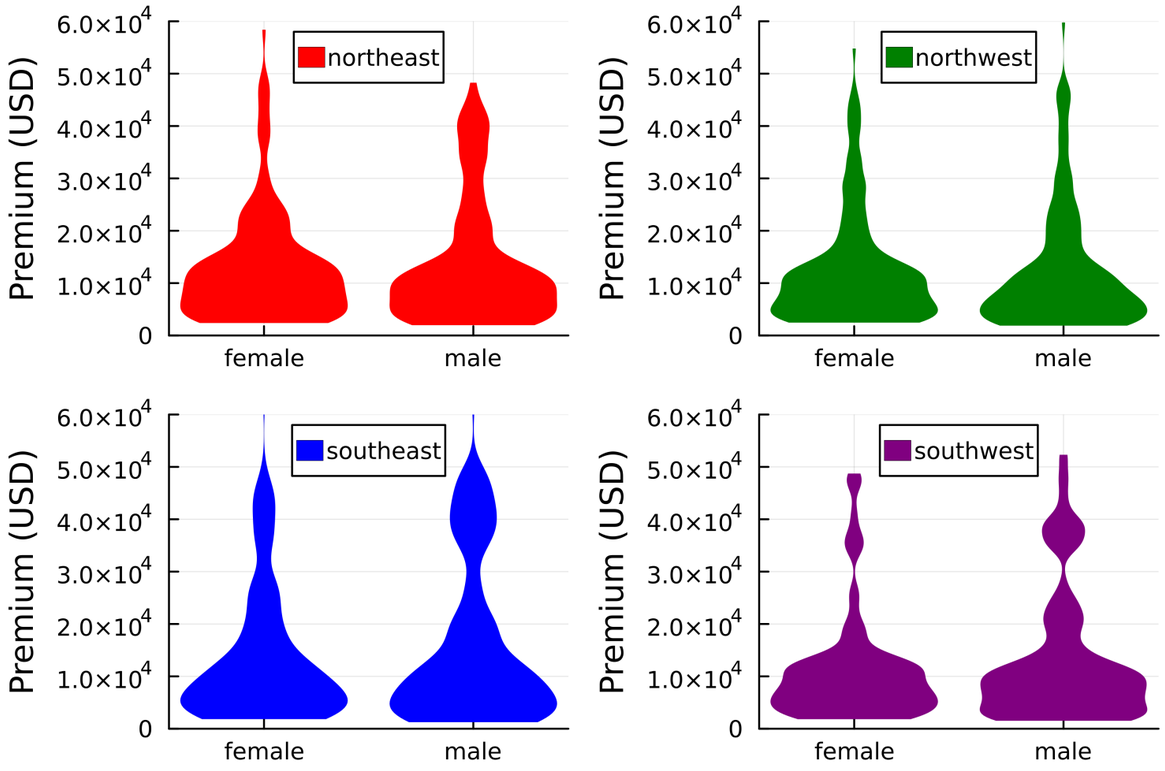
- `insurance` DataFrame

Age	Sex	BMI	Children	Smoker	Region	Charges
19	female	27.90	0	yes	southwest	16884.90
18	male	33.77	1	no	southeast	1725.55
28	male	33.00	3	no	southeast	4449.46
...	...	...	...	...	...	...

- Categorical column → visualize side-by-side plots
  - `@df` recipe is compatible with the `layout` argument!

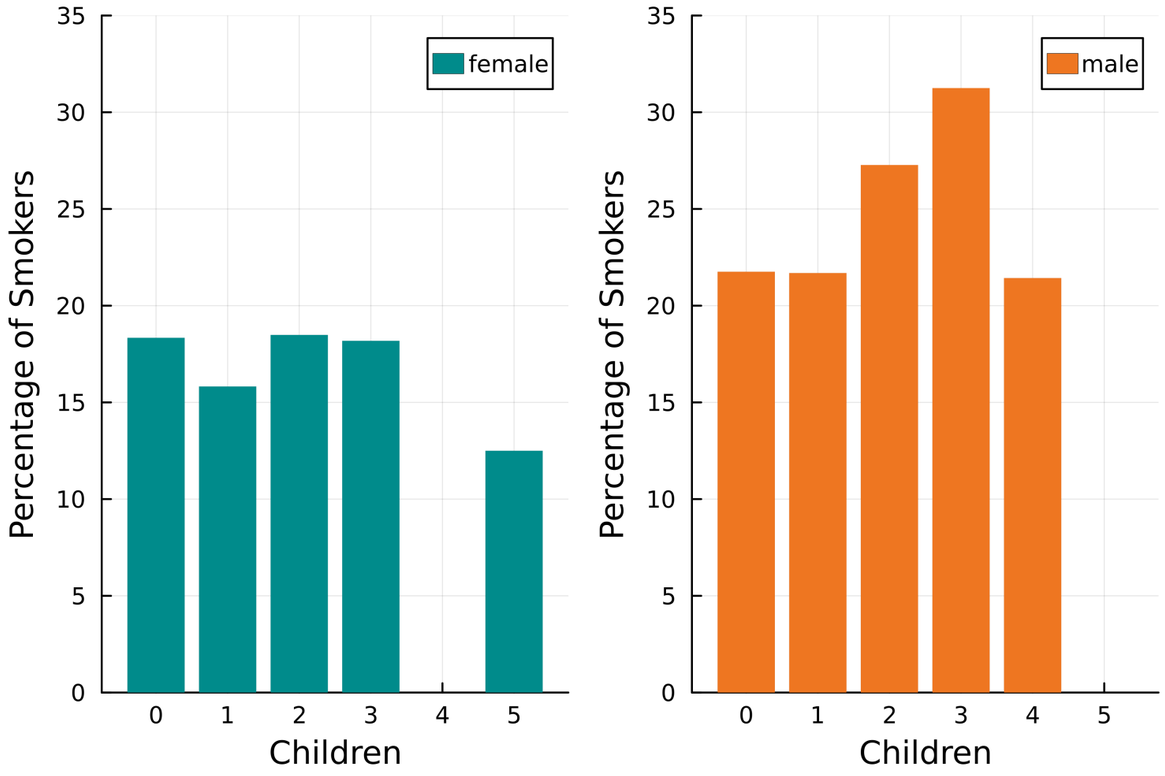
Layouts with DataFrames

```
@df insurance violin(
    :Sex,
    :Charges,
    group=:Region,
    linewidth=0,
    color=[:red :green :blue :purple],
    legend_position=:top,
    # Set layout
    layout=(2,2)
)
ylims!(0, 6*10^4)
ylabel!("Premium (USD)")
```

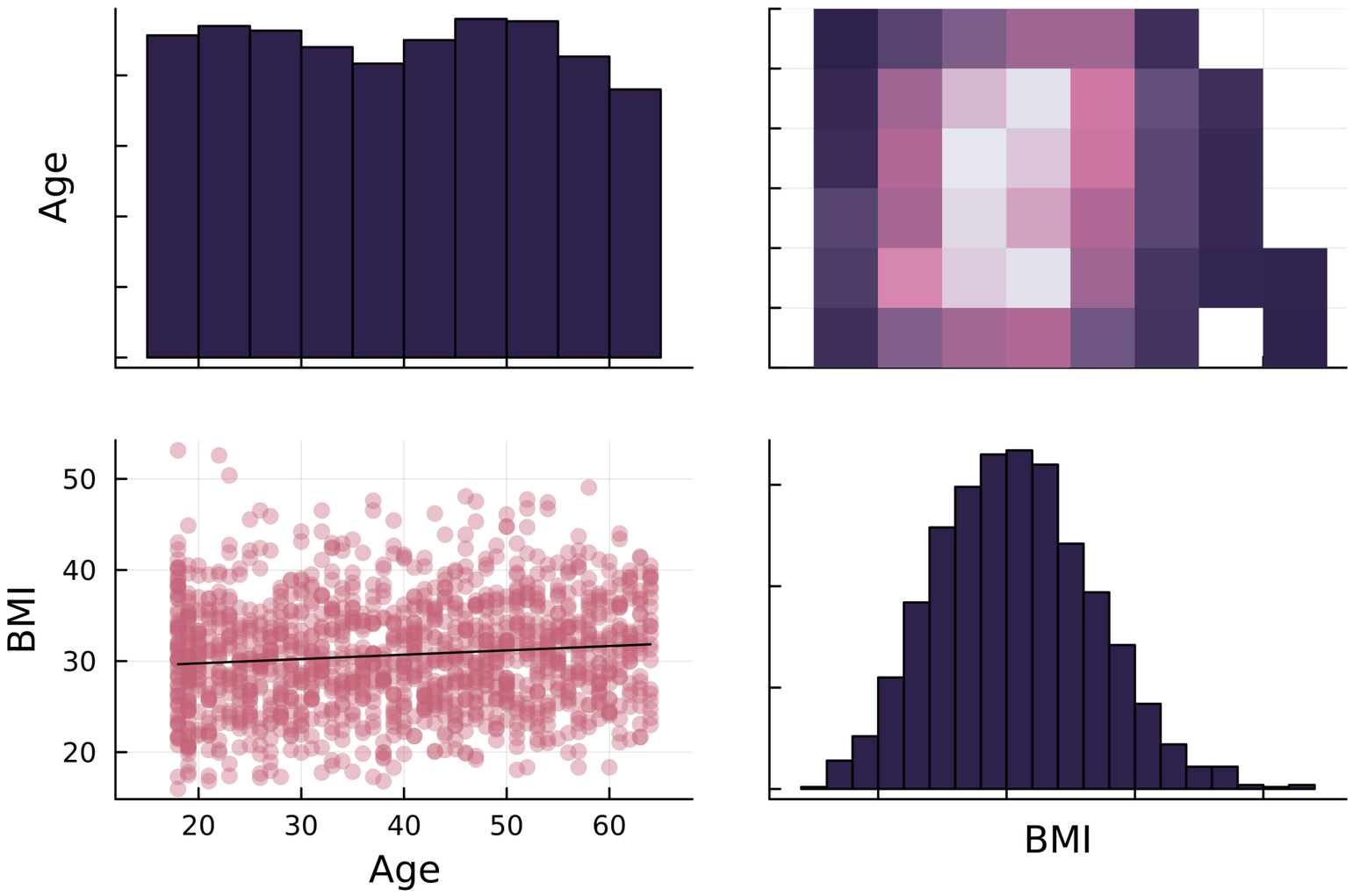


Adding chains to the mix

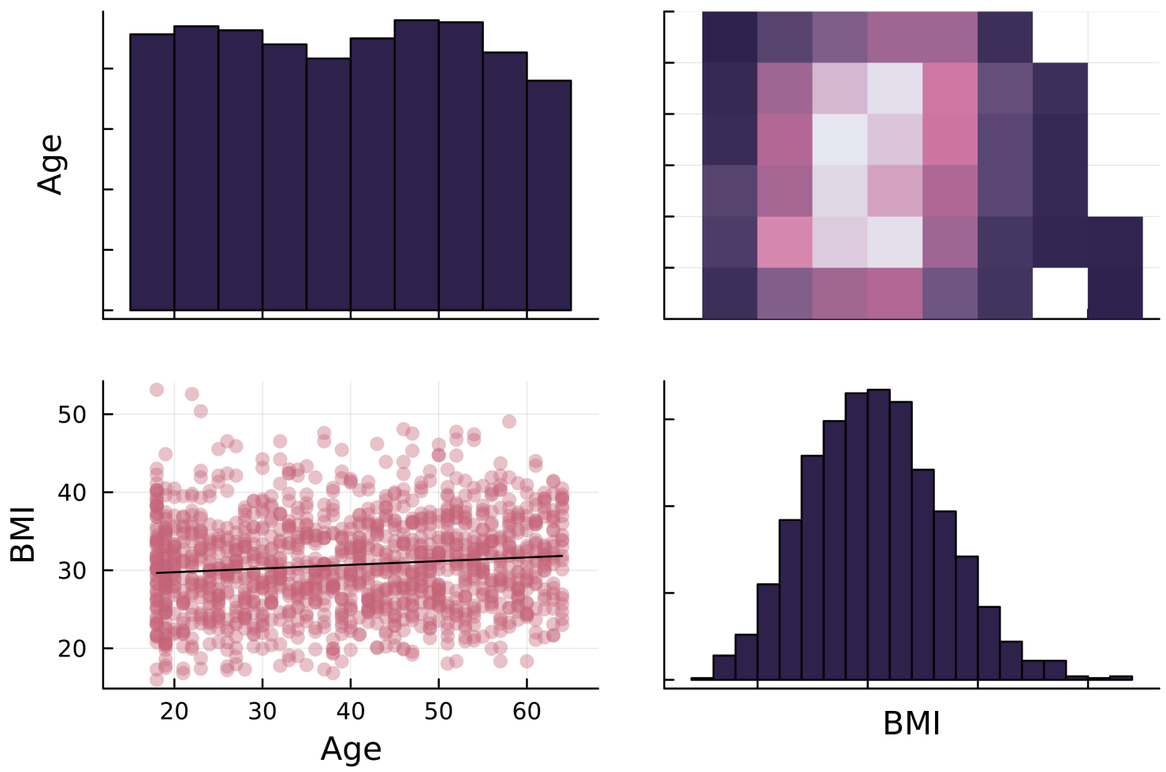
```
@chain insurance begin
    # Smoker column to numeric
    transform(:Smoker
        => ByRow(x -> x == "yes" ? 100 : 0)
        => :Smoker)
    groupby([:Sex, :Children])
    combine(:Smoker => mean)
    @df bar(:Children, :Smoker_mean,
        group=:Sex, linewidth=0,
        fillcolor=[:cyan4 :chocolate2],
        # Set layout
        layout=2)
end
ylims!(0, 35)
xlabel!("Children")
ylabel!("Percentage of Smokers")
```



Correlation matrix plots



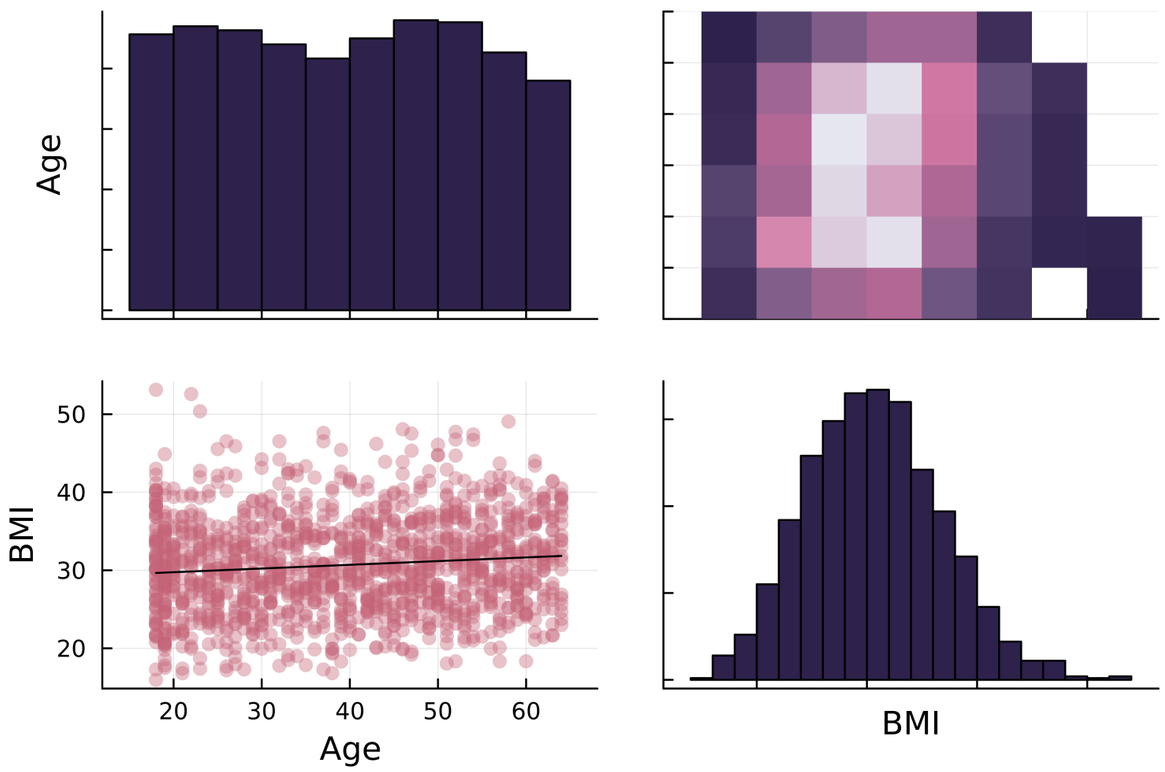
Correlation matrix plots



- Diagonals
  - Histograms of variable distributions
- Above diagonal
  - Two-dimensional histograms
- Below diagonal
  - Scatter plots with regression lines

Correlation matrix plots in StatsPlots.jl

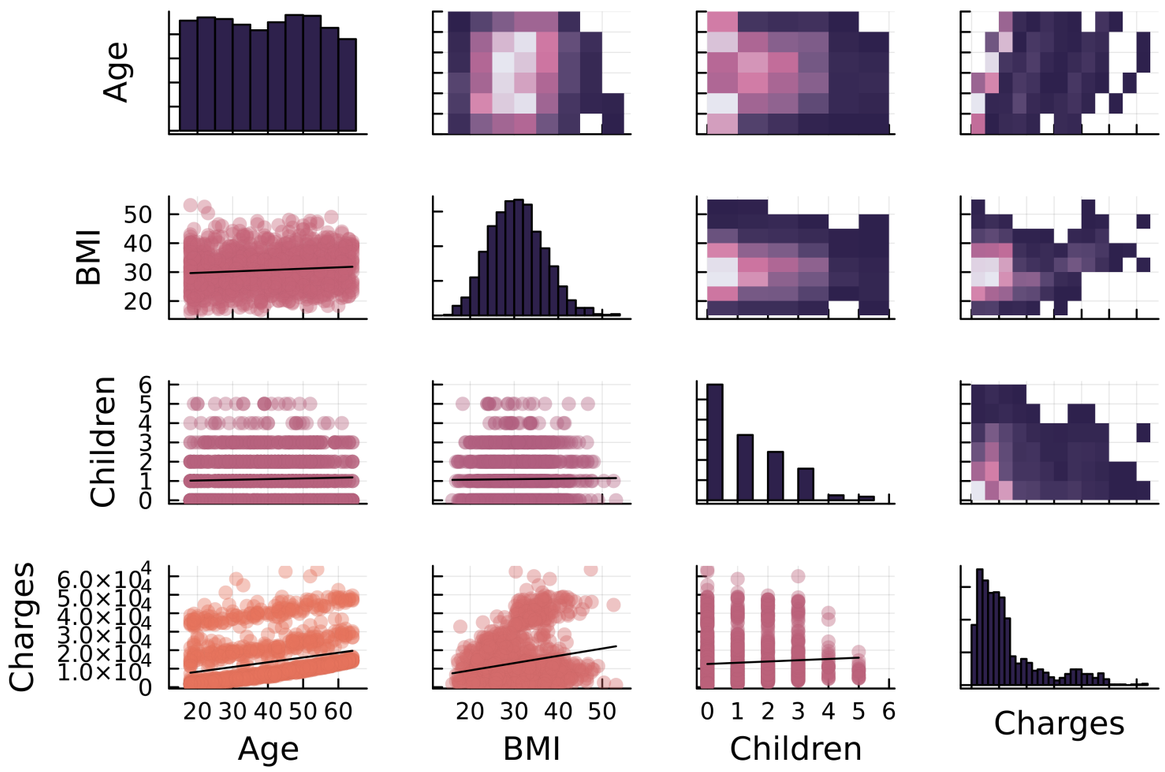
```julia
# Using DataFrames recipe
@df insurance corrplot(
    # Numerical columns
    [:Age :BMI],
    # Customize
    markercolor=:thermal,
    fillcolor=:acton
)
```



[1] https://docs.juliaplots.org/latest/generated/colorschemes/

Correlation matrix plots in StatsPlots.jl

```julia
# Using DataFrames recipe
@df insurance corrplot(
    # Numerical columns
    [:Age :BMI :Children :Charges],
    # Customize
    markercolor=:thermal,
    fillcolor=:acton
)
```



Let's practice!

INTRODUCTION TO DATA VISUALIZATION WITH JULIA

Wrap-up

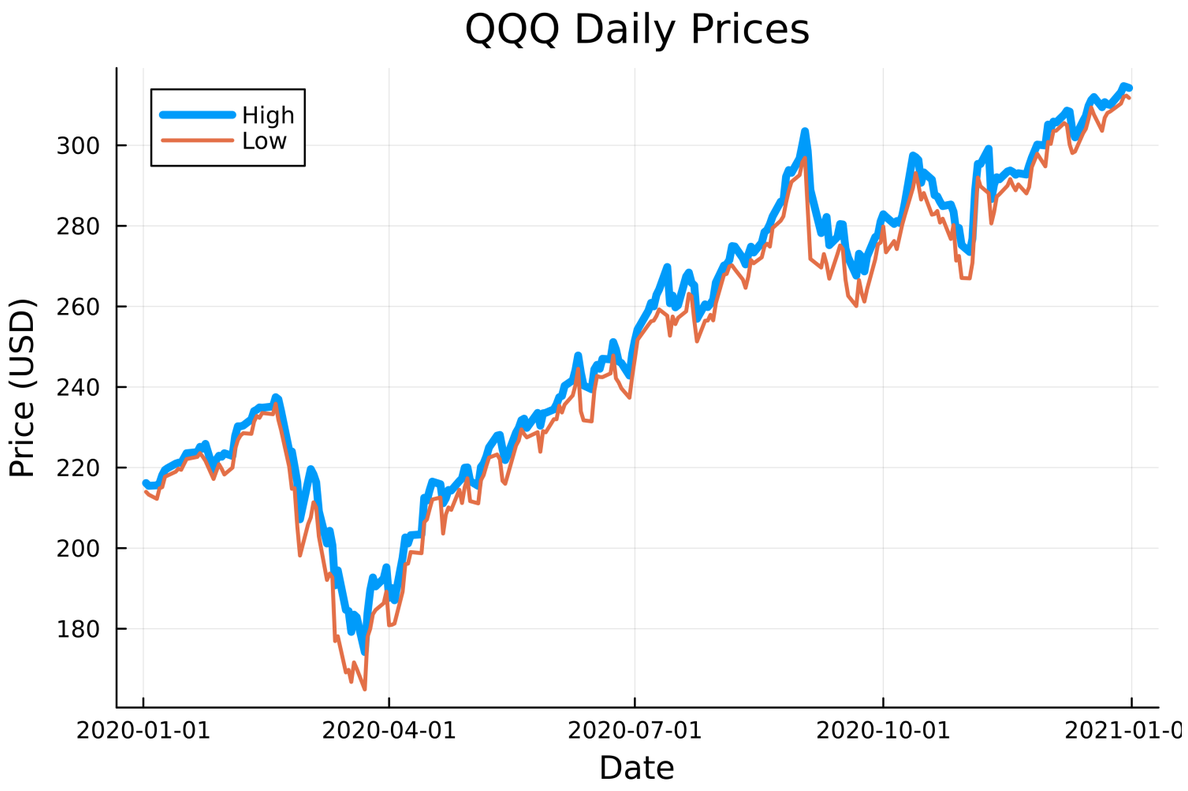
INTRODUCTION TO DATA VISUALIZATION WITH JULIA

Gustavo Vieira Suñe
Data Analyst

datacamp

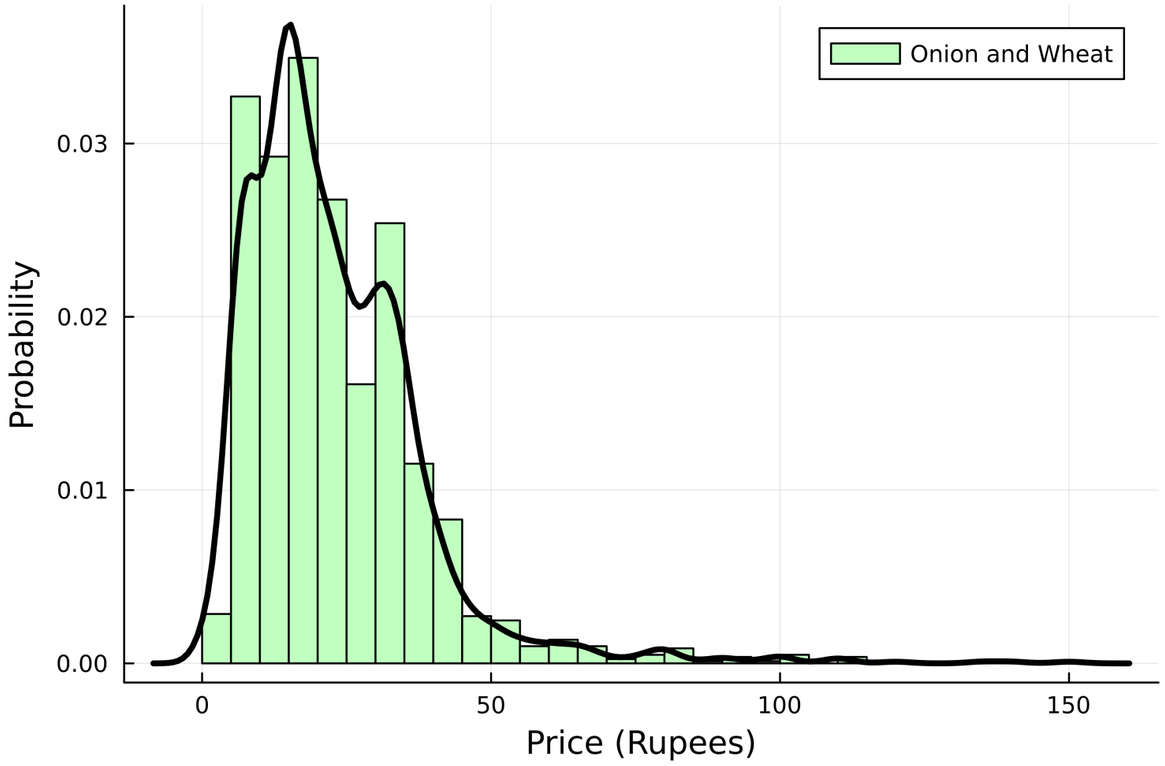
Chapter 1: Basics of Plots.jl

- Why visualize data?
- Plotting with Plots.jl
  - Line plots
  - Scatter plots
- Adding title and axis labels
- Multiple variables in a plot



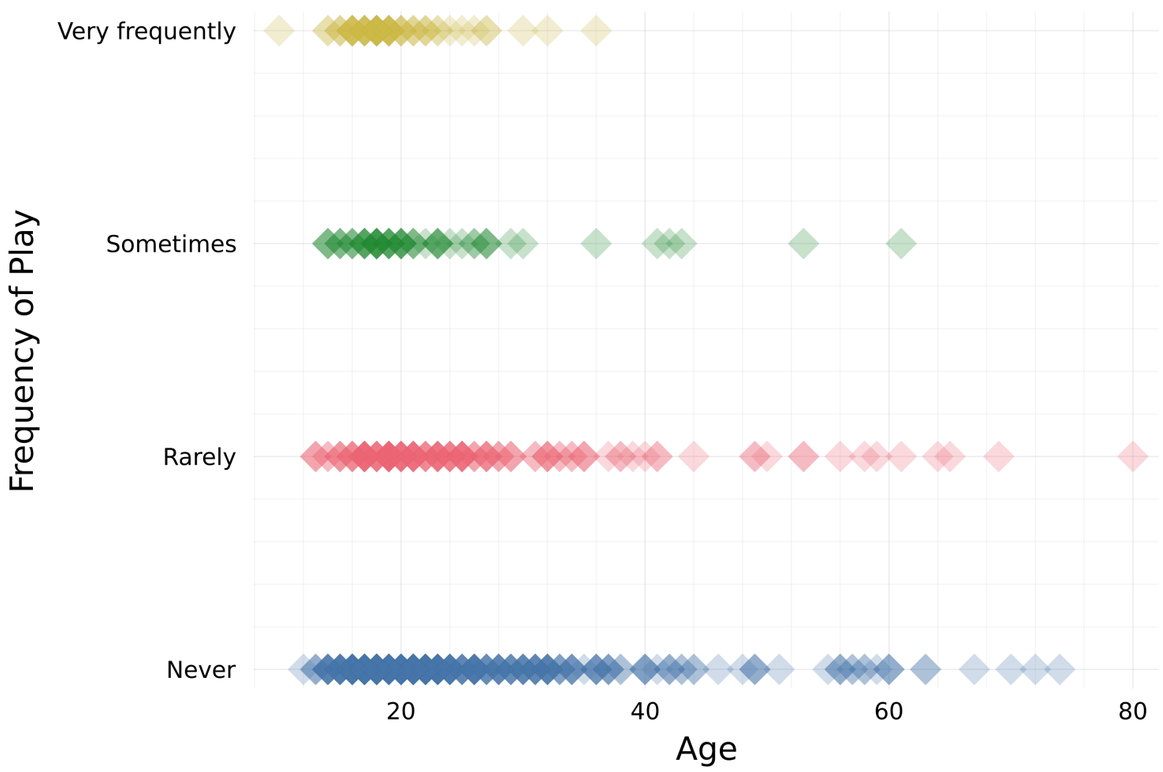
Chapter 2: Advanced plot types

- Advanced plotting with StatsPlots.jl
- Histograms, grouped histograms and density plots
- Bar chars and grouped bar charts
- Box plots and violin plots
- Plotting time series
  - Dealing with dates
- Adding annotations to a plot



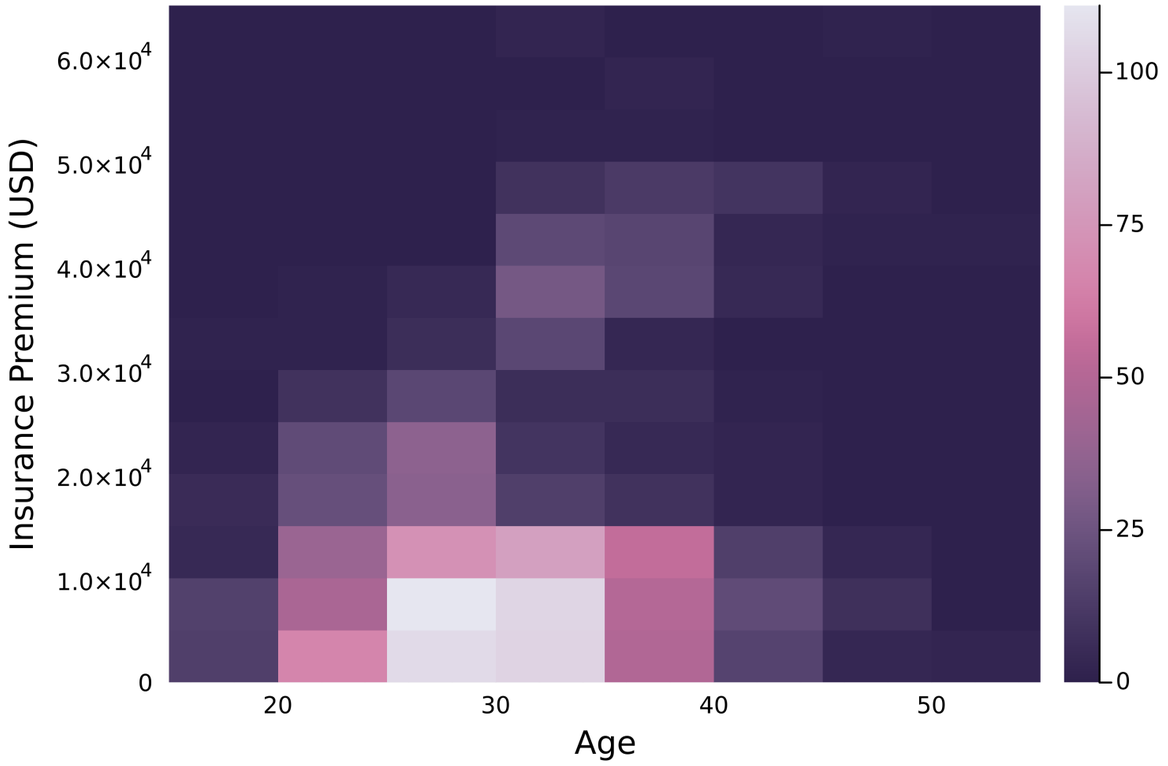
Chapter 3: Mastering customization

- Saving visualizations to file
- Using themes
- Customizing plot attributes
  - Line attributes
  - Marker attributes
  - Axis bounds
  - Legend title and position
  - Transparency
- Layouts
- Series recipes



Chapter 4: Plotting from DataFrames

- Using the StatsPlots.jl DataFrame plotting recipe
- Creating plotting chains with Chain.jl
- Plotting data in more dimensions
  - Three-dimensional scatter plots
  - Two-dimensional histograms
- Layouts and DataFrames
- Correlation matrix plots



Next steps

- Understanding Data Visualization
- Communicating Data Insights
- Dashboard Design Concepts
- Book recommendation: Interactive Visualization and Plotting with Julia

Congratulations!

INTRODUCTION TO DATA VISUALIZATION WITH JULIA