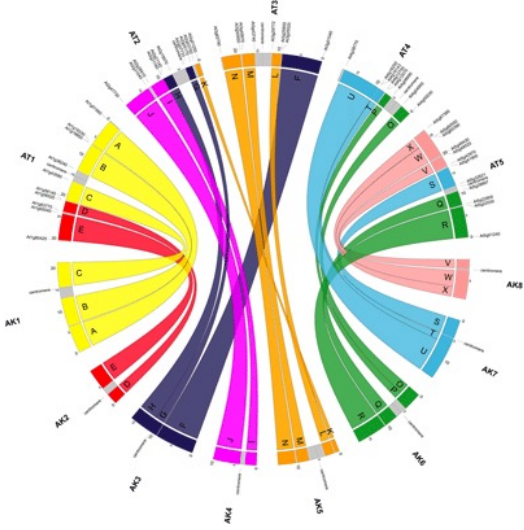
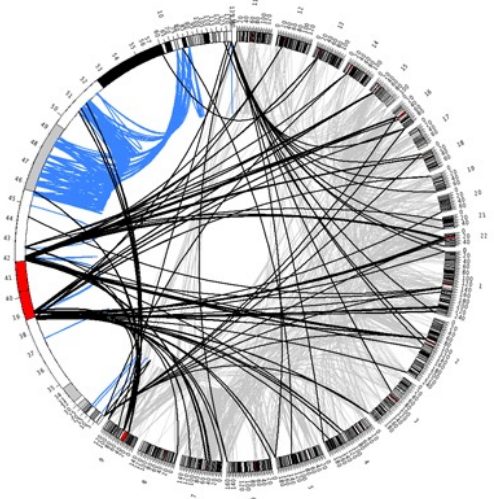
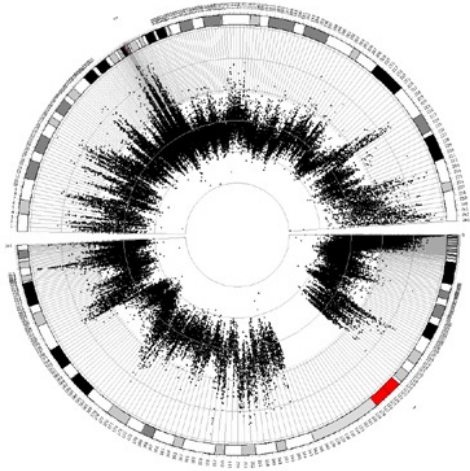
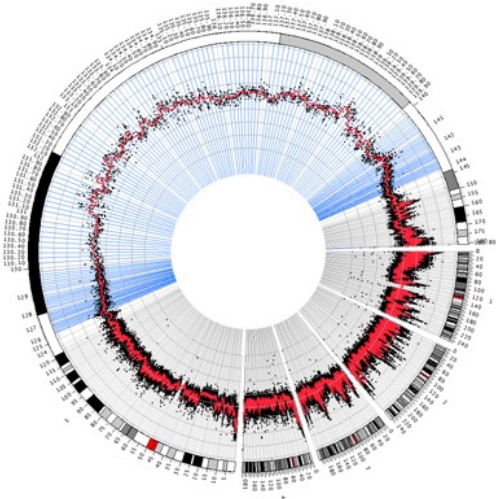


# Data Visualization



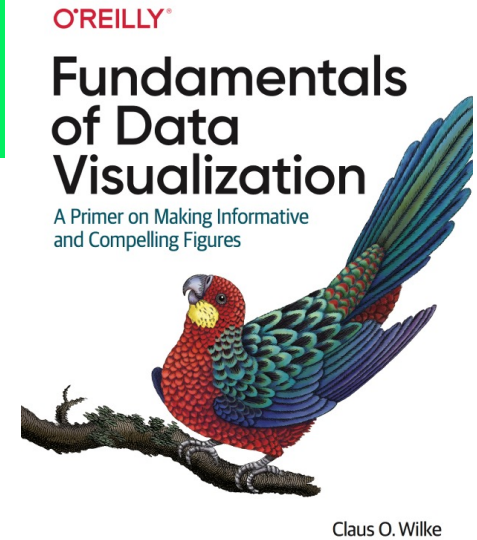
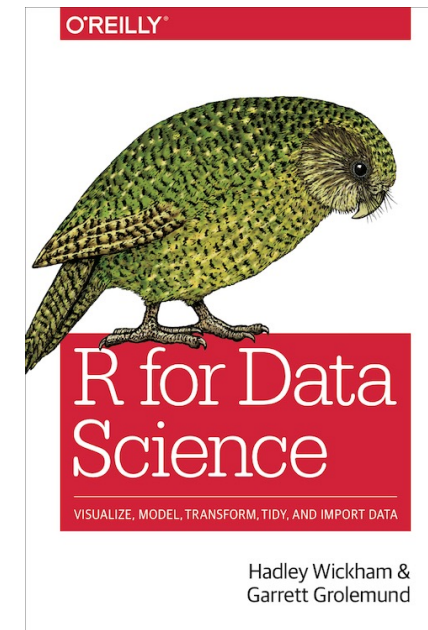
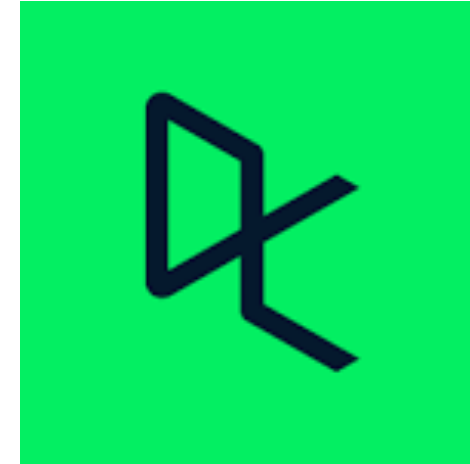
# Data Visualization Resources

## Datacamp

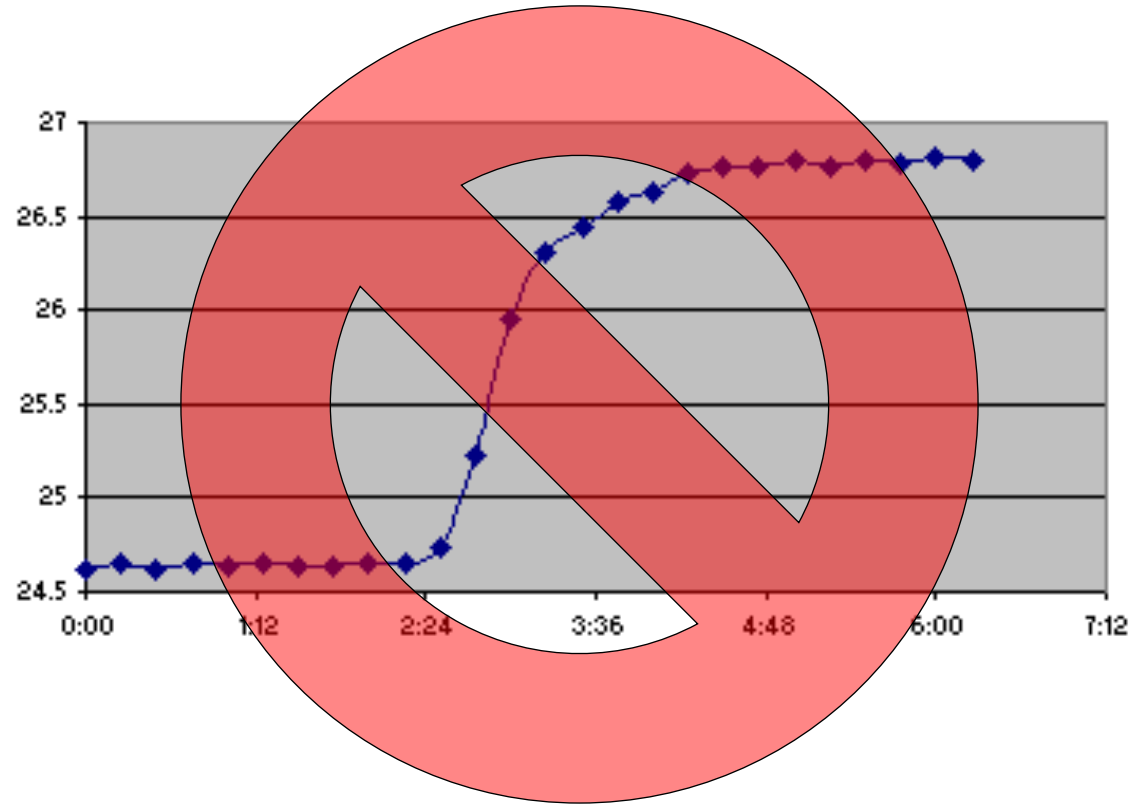
- [Introduction to Data Visualization with ggplot2](#)
- [Intermediate Data Visualization with ggplot2](#)
- [Visualization Best Practices in R](#)

## Claus Wilke Data Visualization in R Course (U Texas)

## R for Data Science



# Scientific Communication and Professionalism

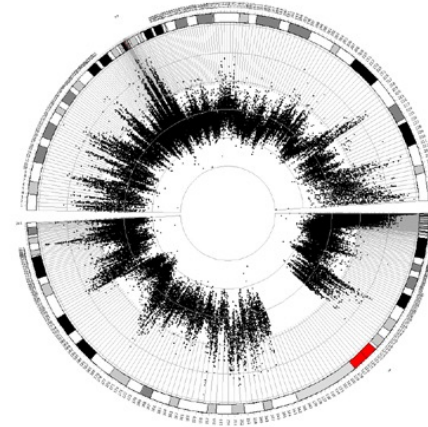


- Clear, accurate, and complete representation of your data
- Efficient, reproducible, and automated methods
- Clean, professional, and aesthetically pleasing appearance

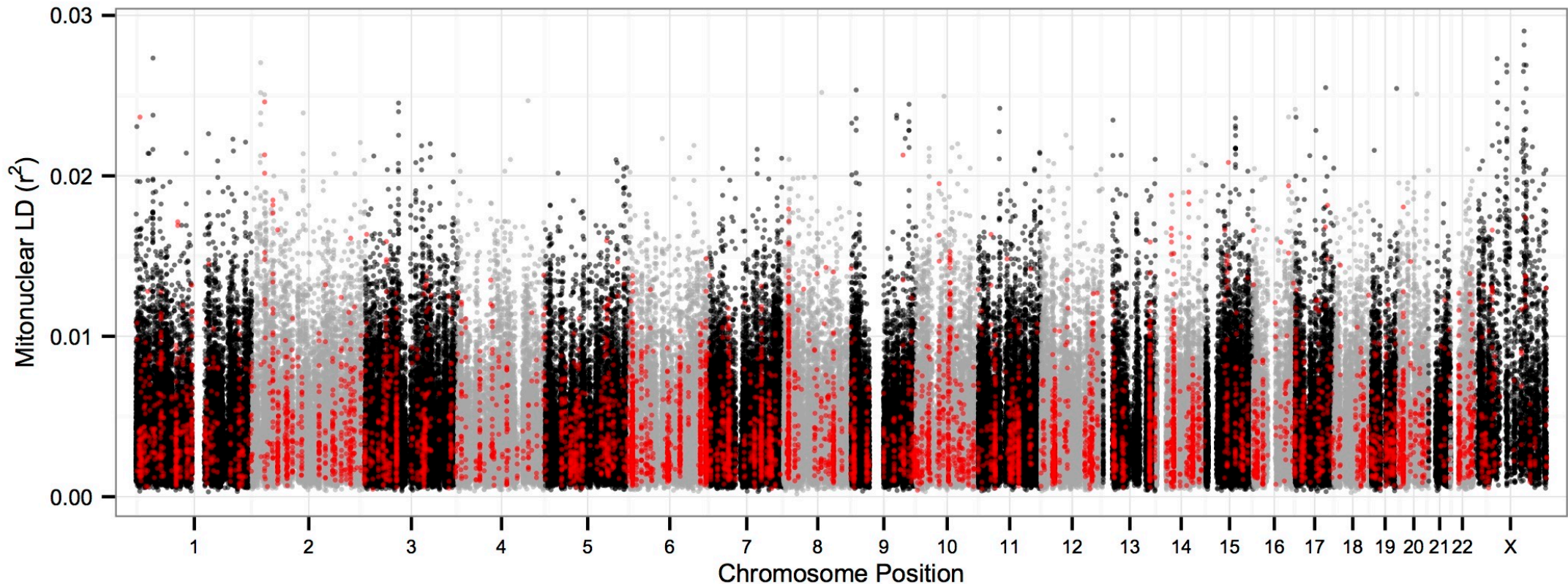
# The Right Tools for the Job

## A few examples

- [R and ggplot](#)
- [Circos](#)
- [Processing](#)
- [Adobe Illustrator](#)



# Making Figures with Code



```
ggplot(cnld) + geom_point(aes(x=CumPos, y=r2, size=0.75, colour=as.factor(ChromPrint), alpha =  
1/8)) + scale_size_identity() + theme_bw(base_size=15) +  
scale_color_manual(values=c(rep(c('black', 'dark gray'),11), 'black', 'red')) +  
scale_x_continuous(expand = c(0.015, 0.015), labels=c(as.character(1:chrNum), "X"),  
breaks=bpMidVec) + theme(plot.margin = unit( c(0.03,0.03,0.03,0.03) , "in" ),  
legend.position='none', axis.text.x = element_text(size=6), axis.text.y = element_text(size=7),  
axis.title.x = element_text(size=8), axis.title.y = element_text(size=8)) + xlab('Chromosome  
Position') + ylab(expression(paste("Mitonuclear LD (",r^2, ")")))
```

**[https://dbsloan.github.io/TS2022/exercises/r\\_markdown.html](https://dbsloan.github.io/TS2022/exercises/r_markdown.html)**

# The Grammar of Graphics

- **aes**: Aesthetic mapping of data to plot elements
  - position (X or Y coordinates), shapes, sizes, color, line weight/type, transparency, etc.
- **geoms**: Layers visually representing your mapped data
  - points, lines, bars, density curves, etc.
- **themes**: Non-data plot elements
  - axis labels, grid lines, titles, etc.

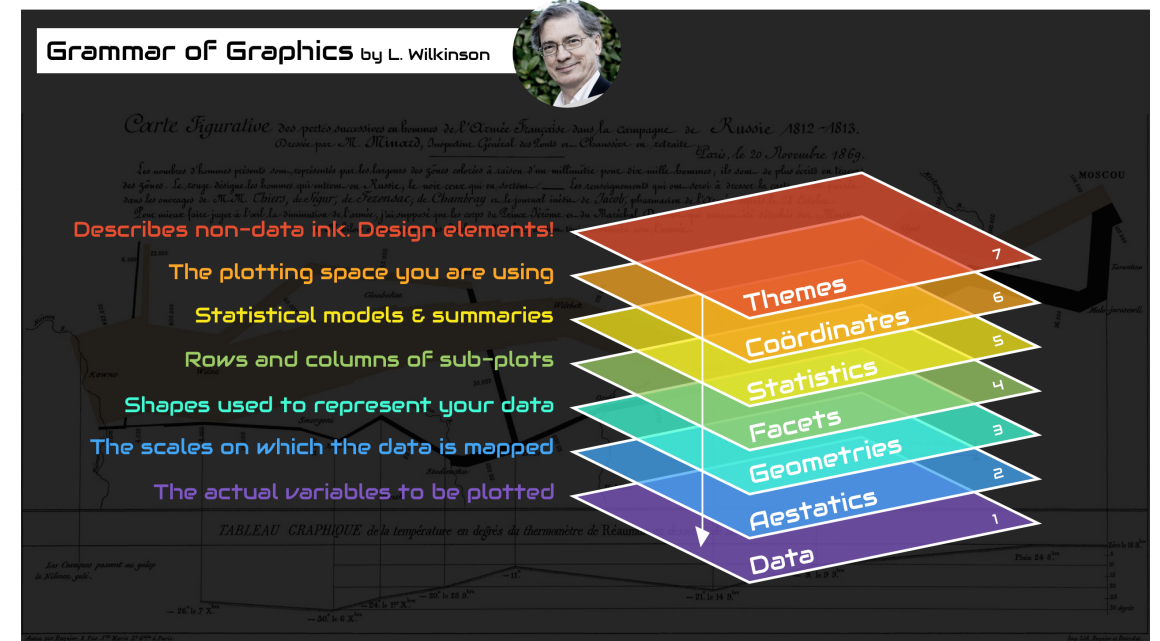


Image: Thomas de Beus

# A Few Principles of Data Visualization

- The Demise of the Bar Plot
- Choosing a Scale: Log vs. Linear
- Accessibility for a Diverse Audience



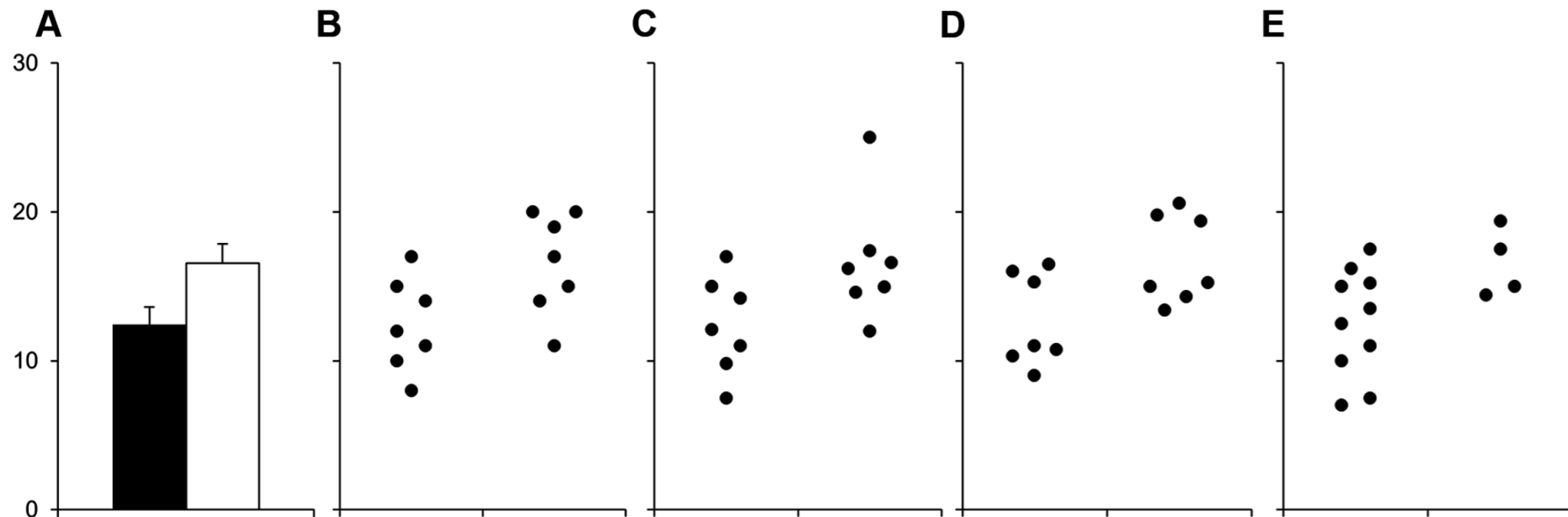
# The Demise of the Bar Plot

PLOS BIOLOGY

## Beyond Bar and Line Graphs: Time for a New Data Presentation Paradigm

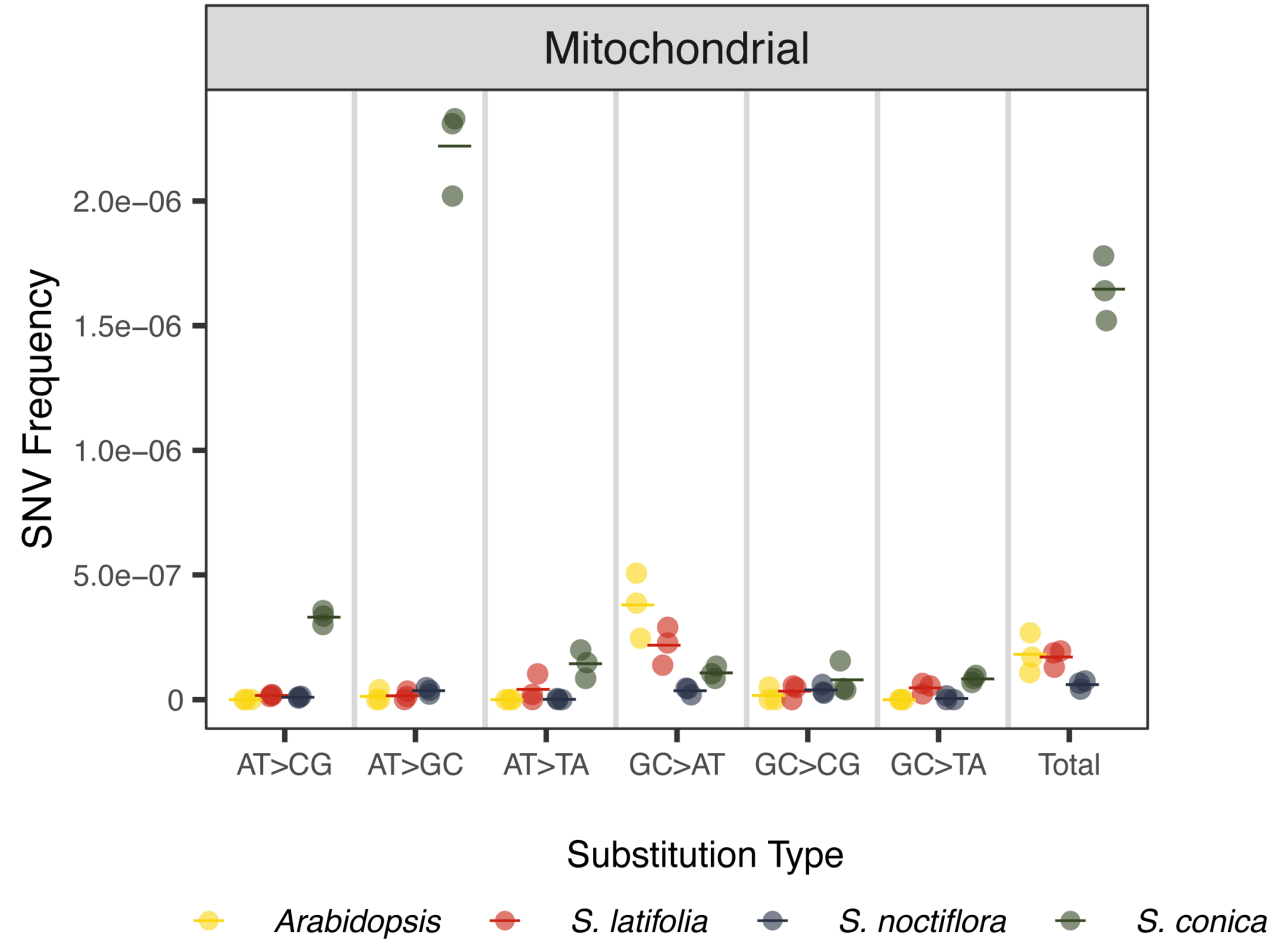
Tracey L. Weissgerber , Natasa M. Milic, Stacey J. Winham, Vesna D. Garovic

Published: April 22, 2015 • <https://doi.org/10.1371/journal.pbio.1002128>



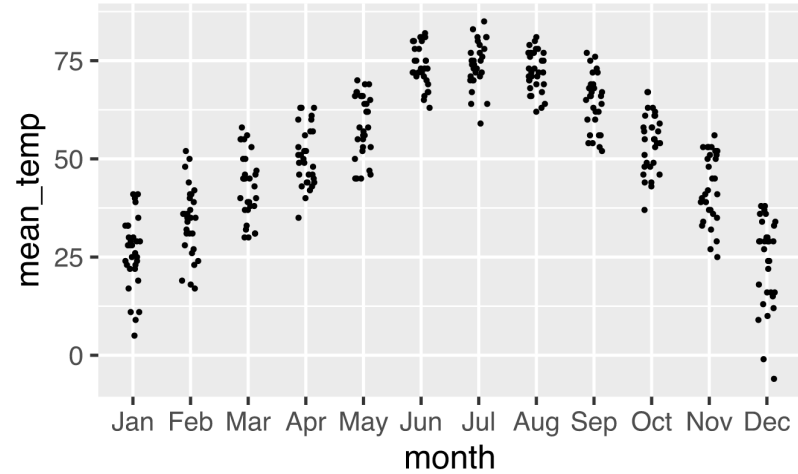
# When Possible... Show All the Data!

Use point size, jitter, and/or transparency to mitigate the effects of overlapping points.

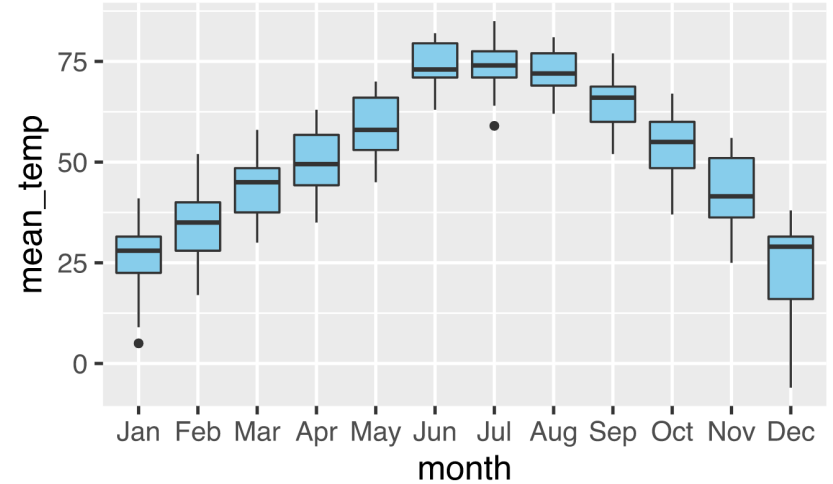


# Better Ways of Comparing and Summarizing Distributions

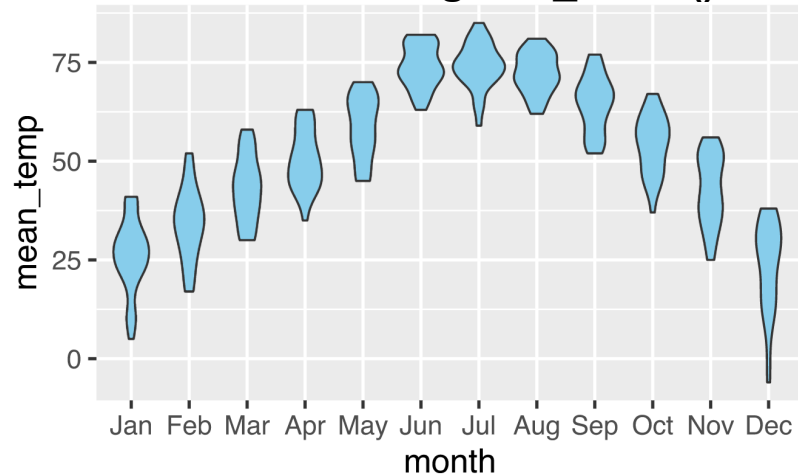
All Data Points: `geom_point()` [with jitter]



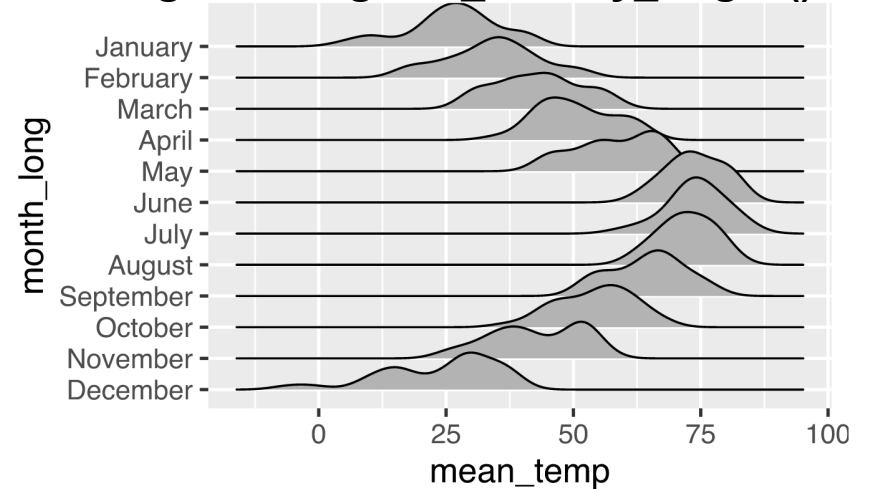
Box Plots: `geom_boxplot()`



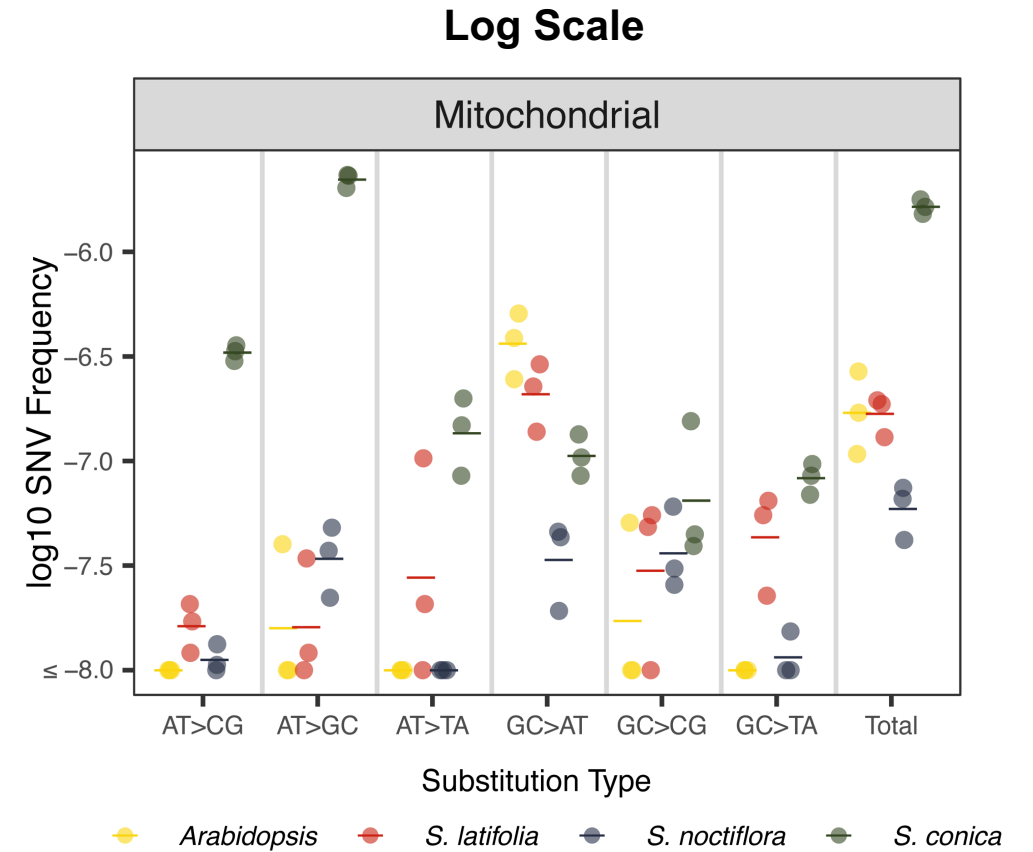
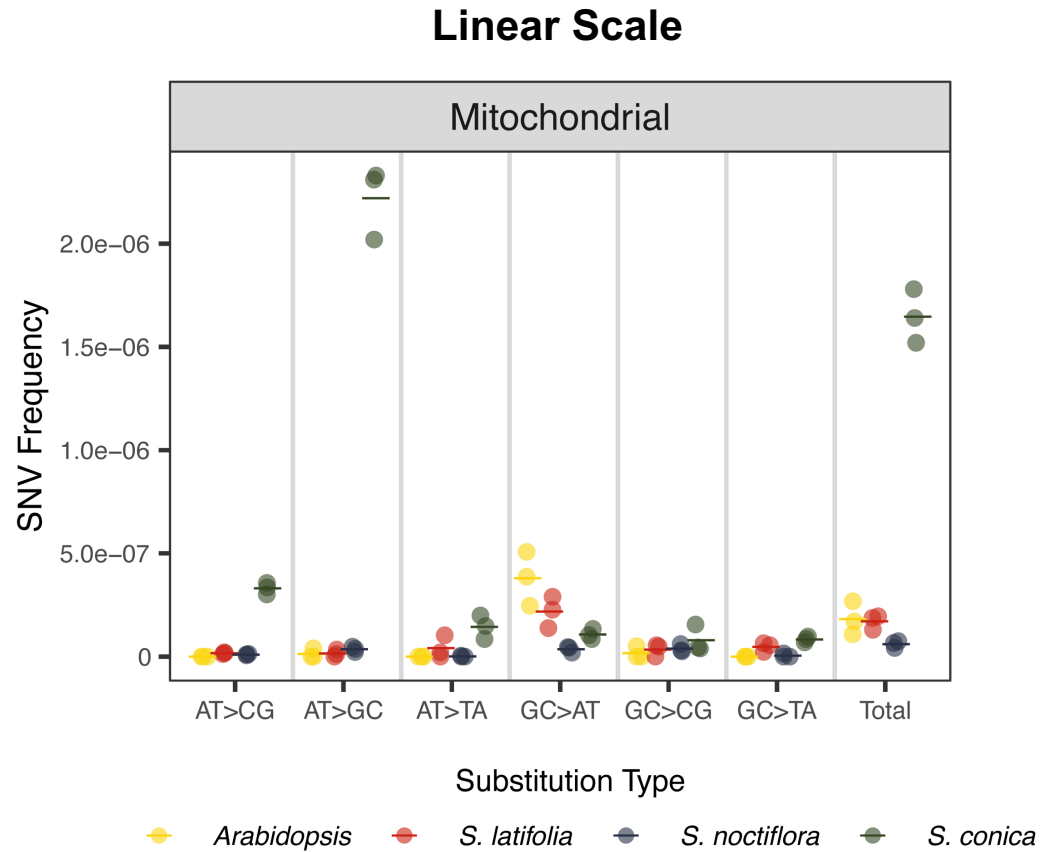
Violin Plots: `geom_violin()`



Ridge Plots: `geom_density_ridges()`



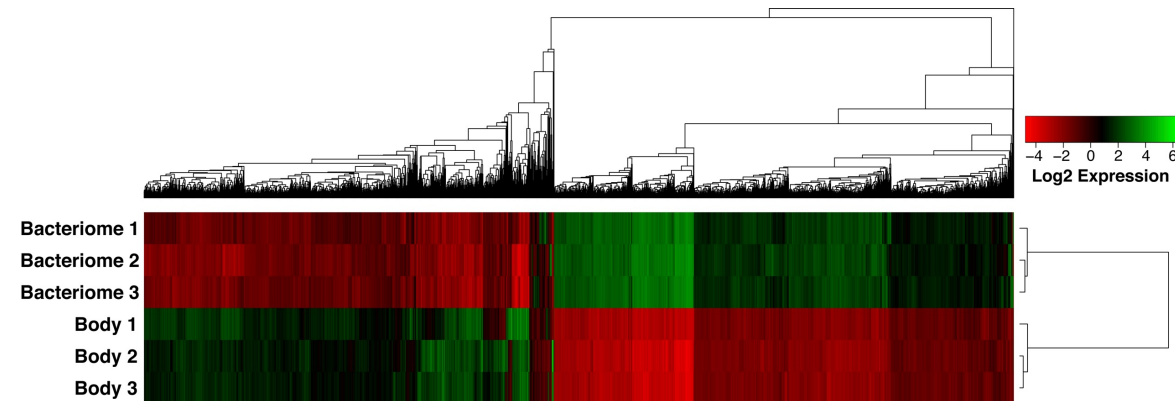
# Linear vs. Log Scales



- Use **linear** scales to emphasize **absolute** differences.
- Use **log** scales to emphasize **proportional** differences.

# Accessibility

Color is a powerful tool for visualizations, but it will not be perceived in the same way by everyone in your audience. Tips for making your visualizations accessible to color blind individuals....



- Use [palettes consisting of colors that are more distinguishable](#) for individuals with common forms of color blindness.
- Use color and shape of points redundantly to distinguish among groups in plot.

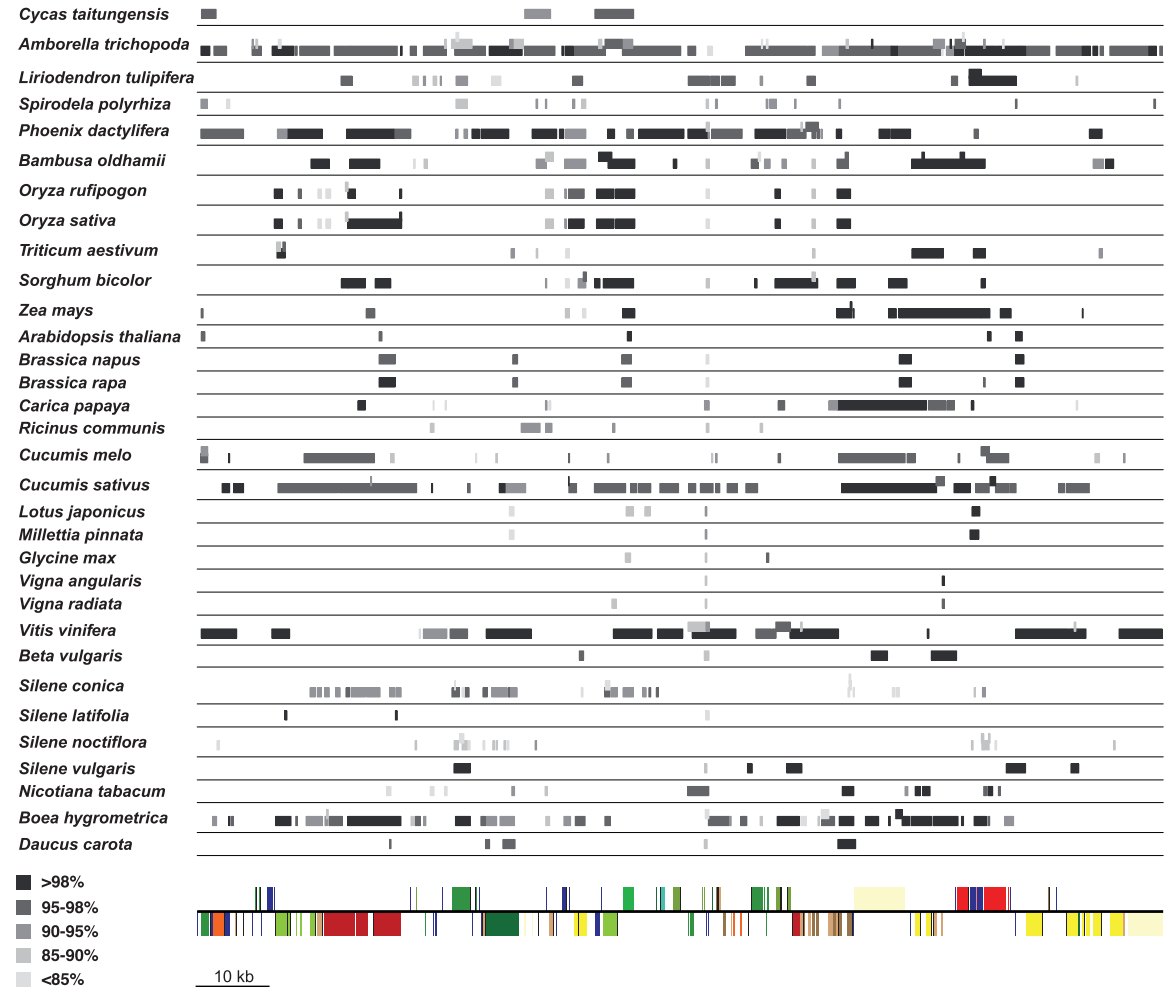
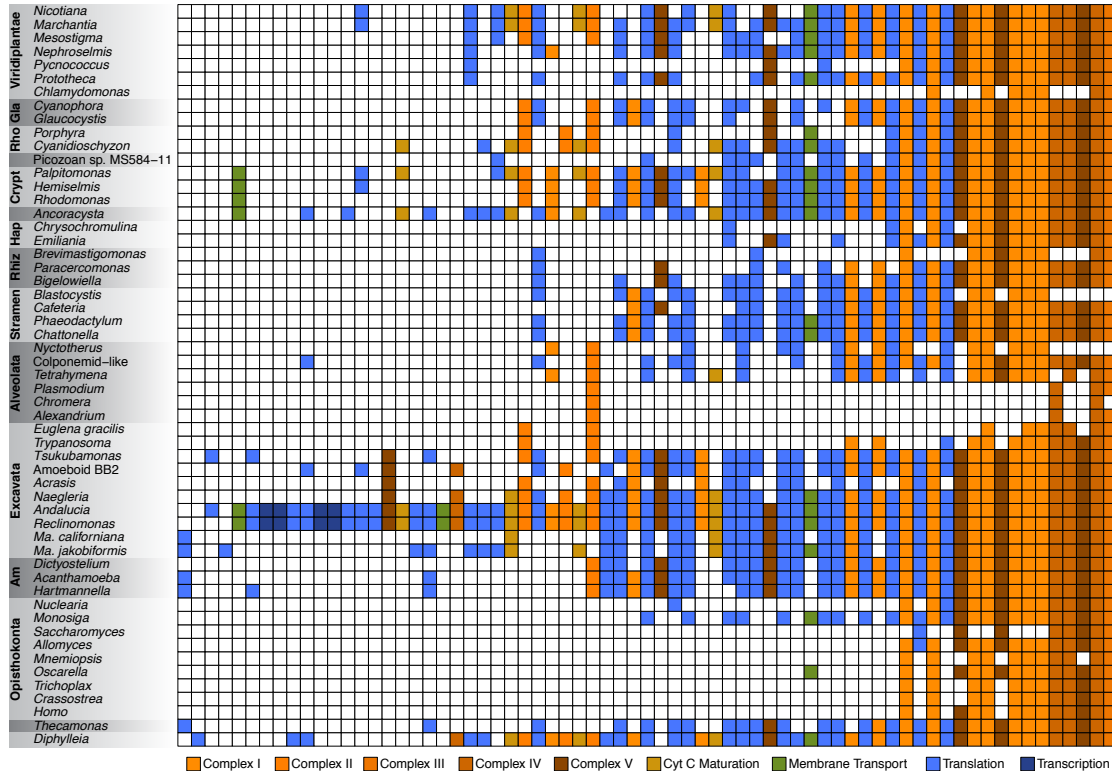
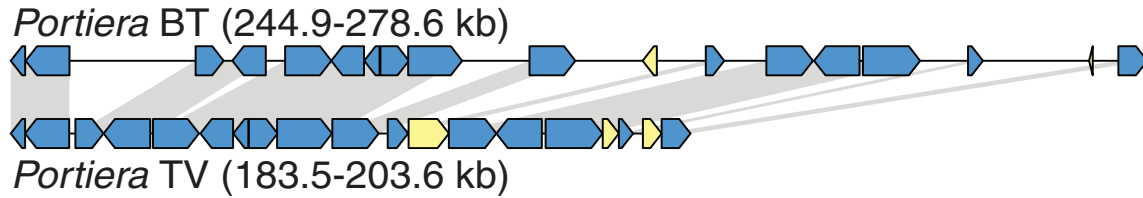
```
> ggplot(MouseData, aes(x=age, y=weight, color=genotype, shape=genotype))  
  + geom_point()
```

- Make use of figure labeling and legend descriptions to make the plot accessible even if colors are difficult to distinguish.

**<https://dbsloan.github.io/TS2022/exercises/ggplot.html>**

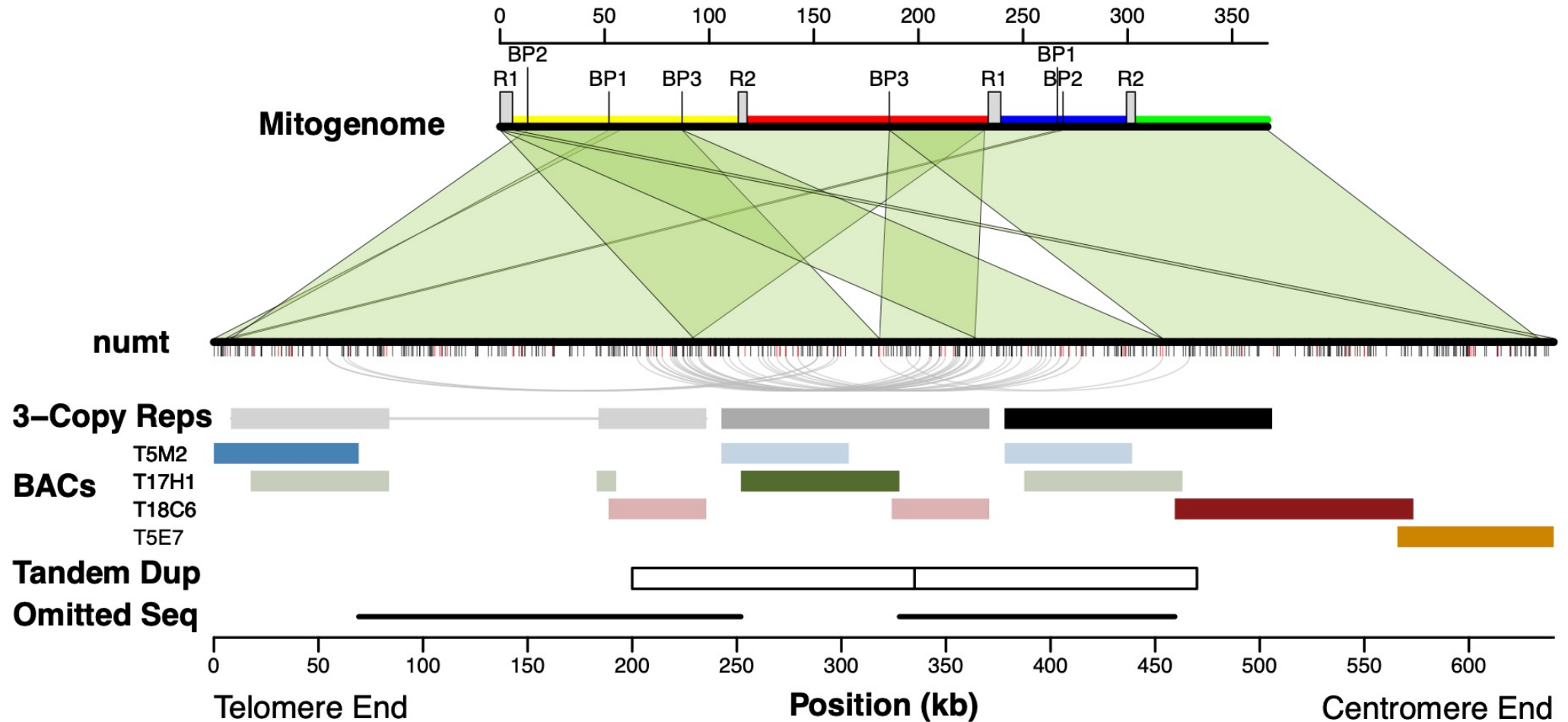
# Drawing with Data

R and other tools can generate custom, data-based visualizations. If you can picture the figure, you can probably generate code to draw it.



# Drawing with Data

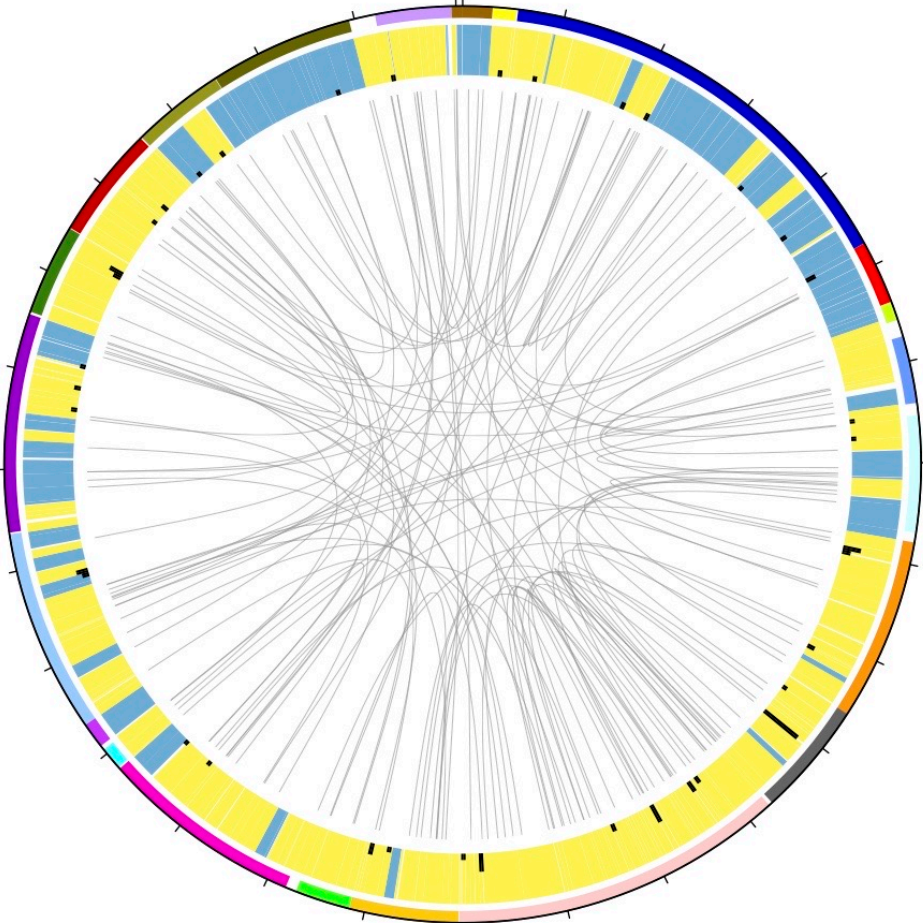
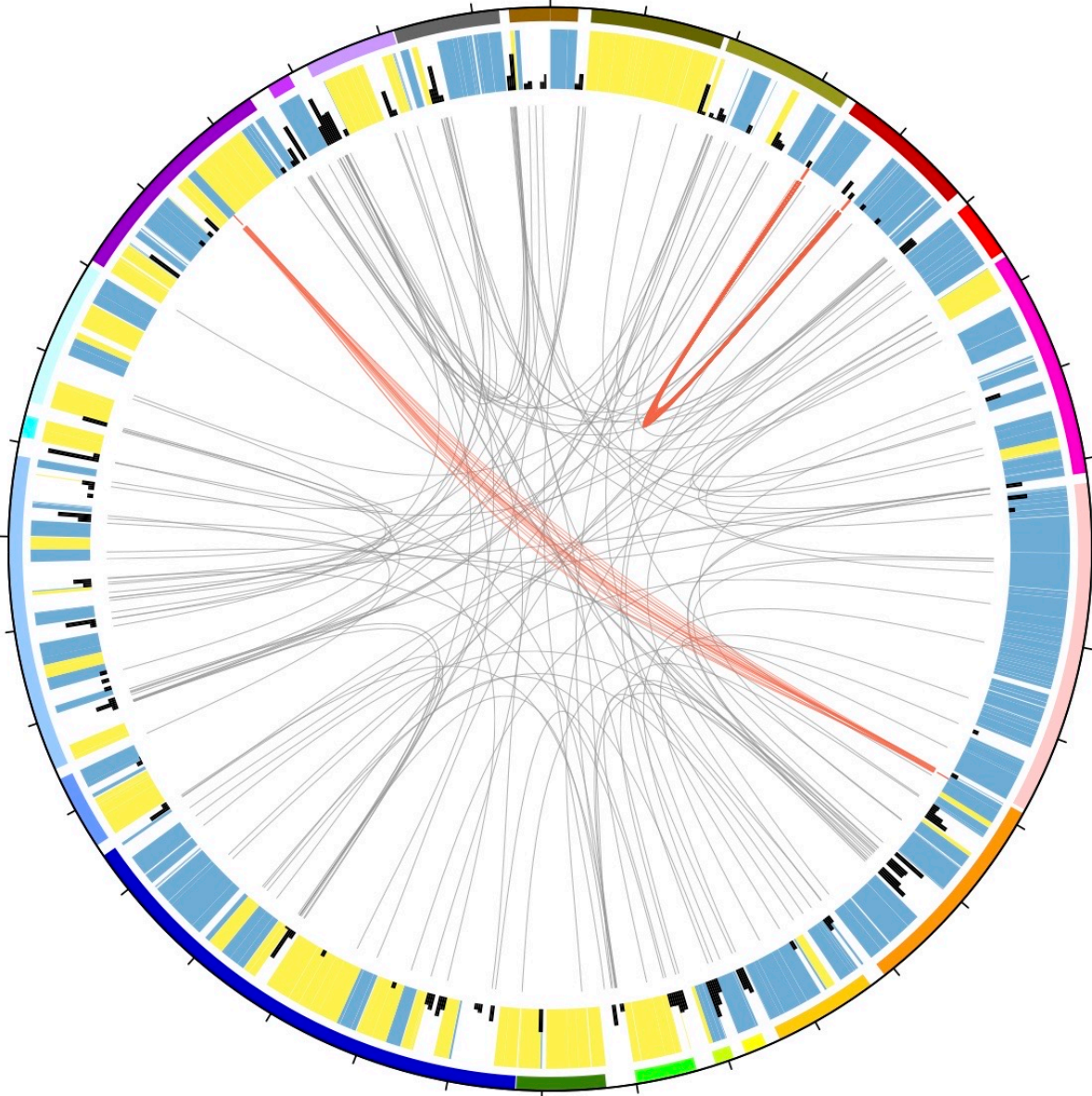
R and other tools can generate custom, data-based visualizations. If you can picture the figure, you can probably generate code to draw it.





**[https://dbsloan.github.io/TS2022/exercises/r\\_figure\\_drawing.html](https://dbsloan.github.io/TS2022/exercises/r_figure_drawing.html)**

# Circular Data Representations



**<https://dbsloan.github.io/TS2022/exercises/circos.html>**