Visualizing Data

Summer Training Academy for Research Success

Michael F. Seese

Department of Political Science
University of California San Diego

Summer 2021

Outline

Replication code is available on GitHub

https://github.com/mfseese/STARS_Summer2021

The Visual Presentation of Data

Why Should We Visualize Data?

1. To make sense of the data
   - 1.1 Identify broad patterns
   - 1.2 Check our assumptions and intuition
2. To tell stories
   - 2.1 Underscore the **puzzle**
   - 2.2 Present the results of our analysis

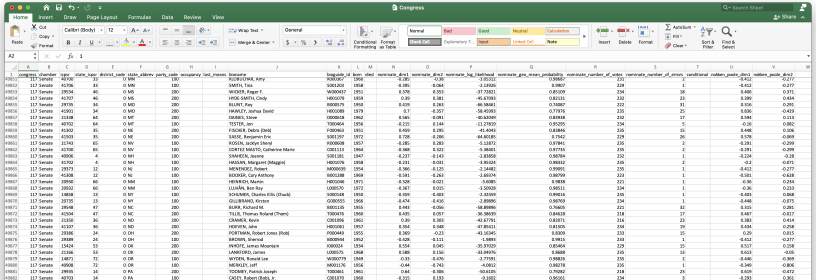
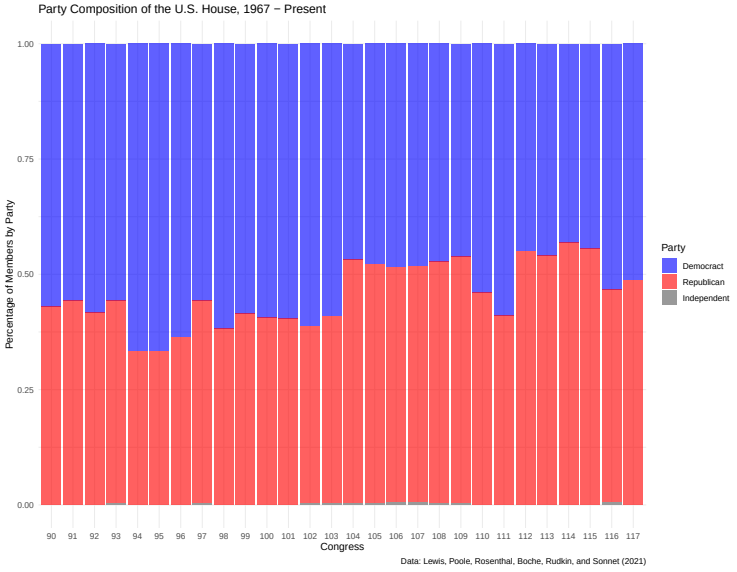
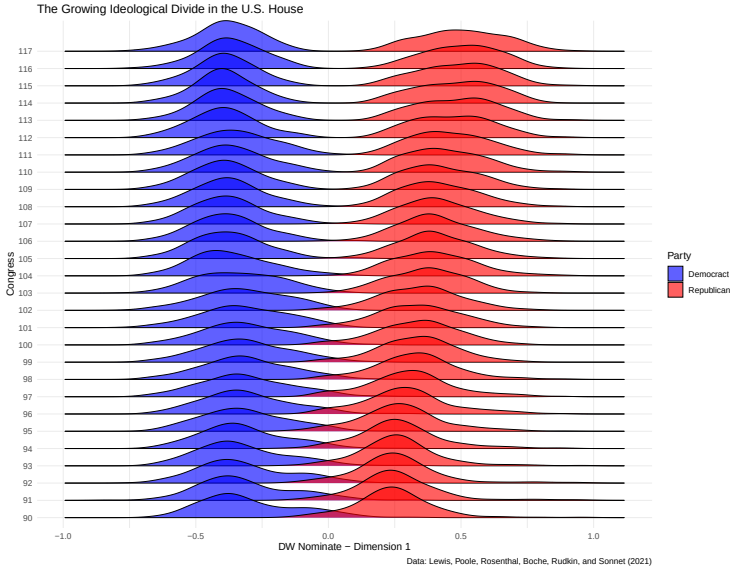
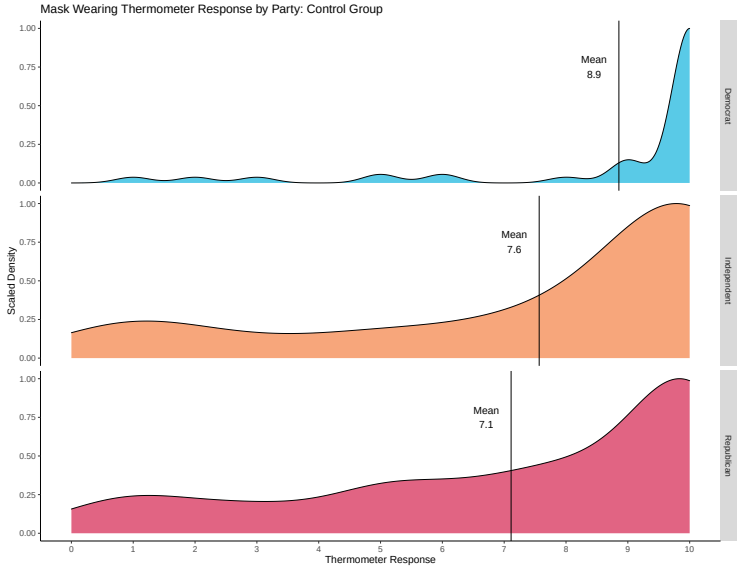
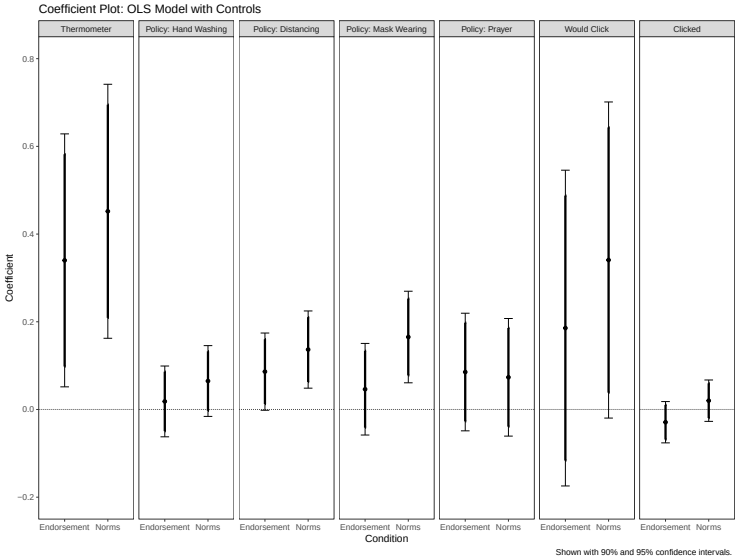


Figure: Member Ideology Data from https://voteview.com/data


Party Composition of the U.S. House, 1967 − Present
Percentage of Members by Party
1.00
0.75
0.50
0.25
0.00
90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117
Congress
Party
Democrat
Republican
Independent
Data: Lewis, Poole, Rosenthal, Boche, Rudkin, and Sonnet (2021)


The Growing Ideological Divide in the U.S. House
Congress
117
116
115
114
113
112
111
110
109
108
107
106
105
104
103
102
101
100
99
98
97
96
95
94
93
92
91
90
−1.0
−0.5
0.0
0.5
1.0
DW Nominate − Dimension 1
Party
Democract
Republican
Data: Lewis, Poole, Rosenthal, Boche, Rudkin, and Sonnet (2021)


Mask Wearing Thermometer Response by Party: Control Group
Scaled Density
1.00
0.75
0.50
0.25
0.00
Mean
8.9
Democrat
Mean
7.6
Independent
Mean
7.1
Republican
0
1
2
3
4
5
6
7
8
9
10
Thermometer Response


Coefficient Plot: OLS Model with Controls
Thermometer
Policy: Hand Washing
Policy: Distancing
Policy: Mask Wearing
Policy: Prayer
Would Click
Clicked
Coefficient
0.8
0.6
0.4
0.2
0.0
−0.2
Endorsement
Norms
Condition
Shown with 90% and 95% confidence intervals.
Note: Model for "Clicked" outcome uses subset data (only petition−eligible respondents).

Worth a Thousand ~~Words~~ Data Points

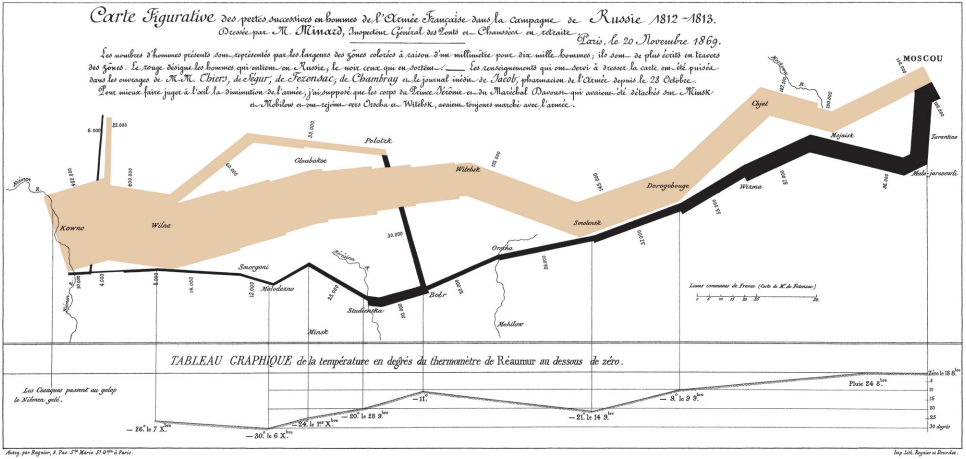


Figure: Minard's *Carte Figurative*, 1869

The Process

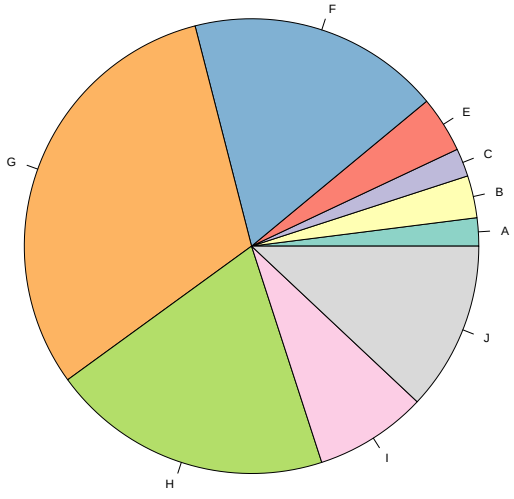
1. Select the appropriate plot type
2. Generate the plot with **proper** formatting
3. Ask yourself (and other people): "Is this legible? Is this intuitive?"
4. Revise with aesthetics in mind
5. Clean your code to ensure replicability

Plot Types

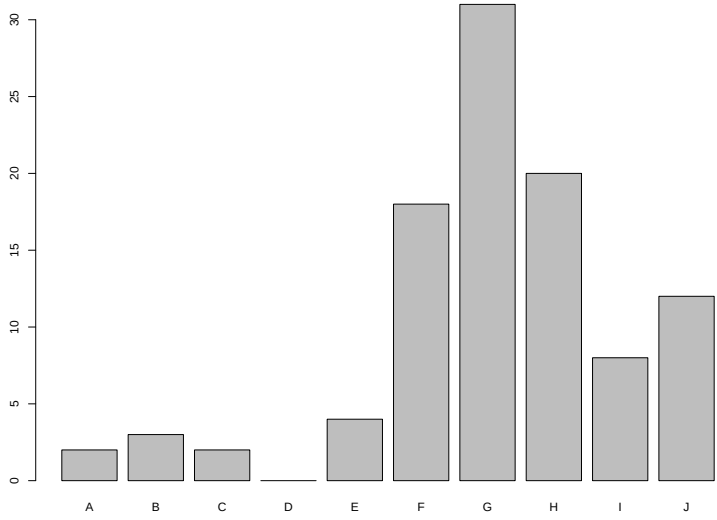
Table: Selecting the Appropriate Plot

1 Variable	$\geqslant$ 2 Variables			
Distributions	Comparison	Relationship	Results	Avoid
Histogram	Bar*	Scatter	Coefficient Plot	Pie
Density Plot	Dot Plot*	Bubble	Margins Plot	Donut
Box Plot	Line	Fit Plots	Diagnostic	Stacked Area
Violin Plot	Time Series			
Thematic Maps	Range Plots			

x	y
A	2
B	3
C	2
D	0
E	4
F	18
G	31
H	20
I	8
J	12

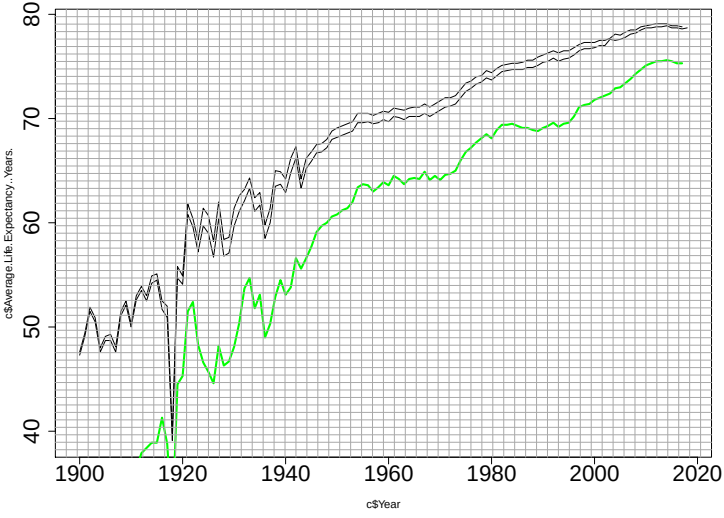


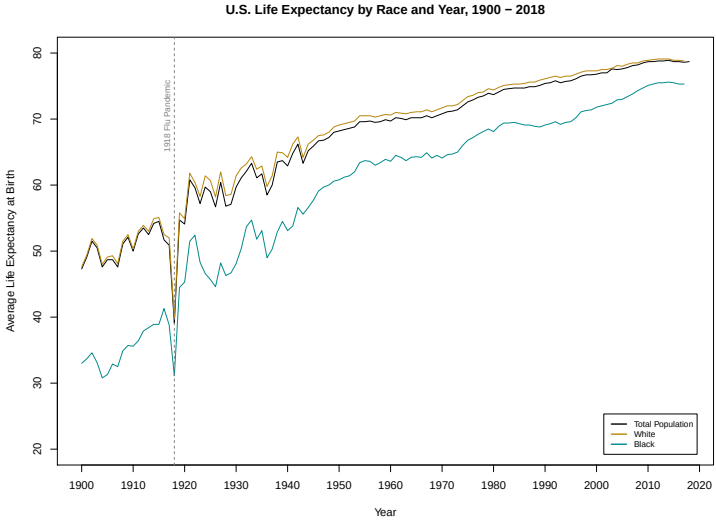
x	y
A	2
B	3
C	2
D	0
E	4
F	18
G	31
H	20
I	8
J	12



Proper Formatting

- Include
  - A (descriptive) title
  - A note or caption with data sources and supplementary information
  - Marker labels where appropriate
  - A complete legend
  - Confidence intervals when possible
- Display
  - All axes and their labels
  - The full range of relevant variation
- Select appropriate color palette
- Always relabel variables for display
- Ensure text is legible (size, direction, etc.)


80
70
60
50
40
c$Average..Life.Expectancy..Years.
1900
1920
1940
1960
1980
2000
2020
c$Year

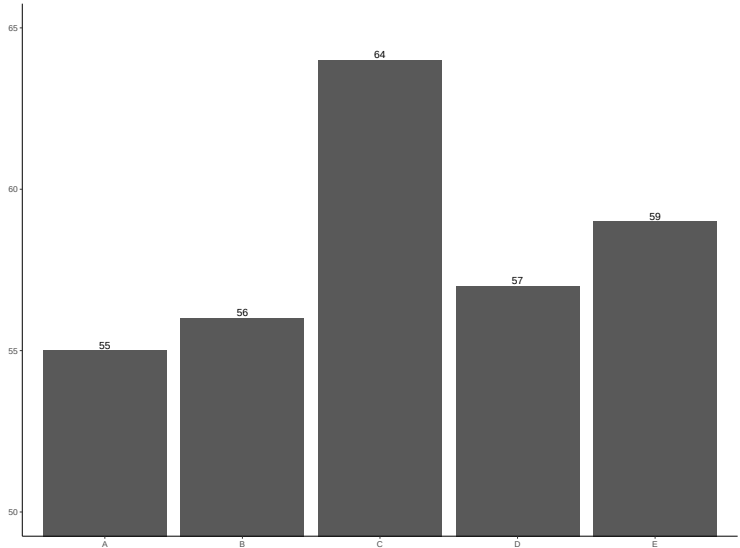


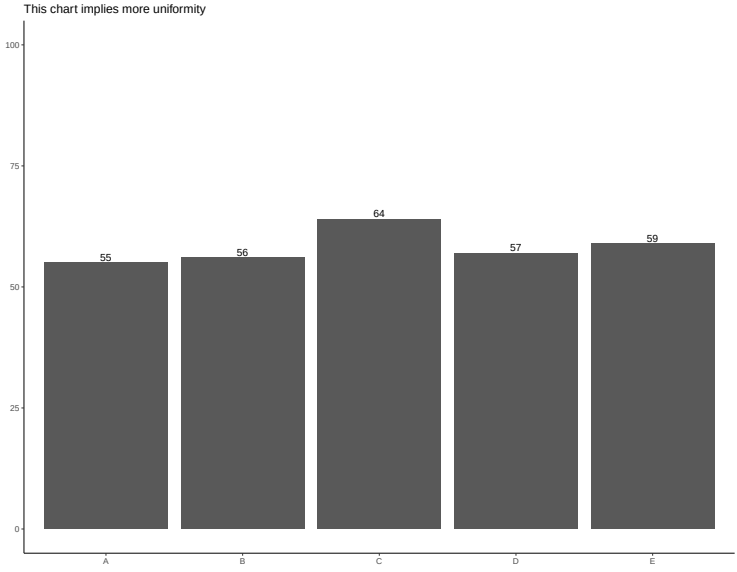
Data: U.S. Centers for Disease Control and Prevention, National Center for Health Statistics

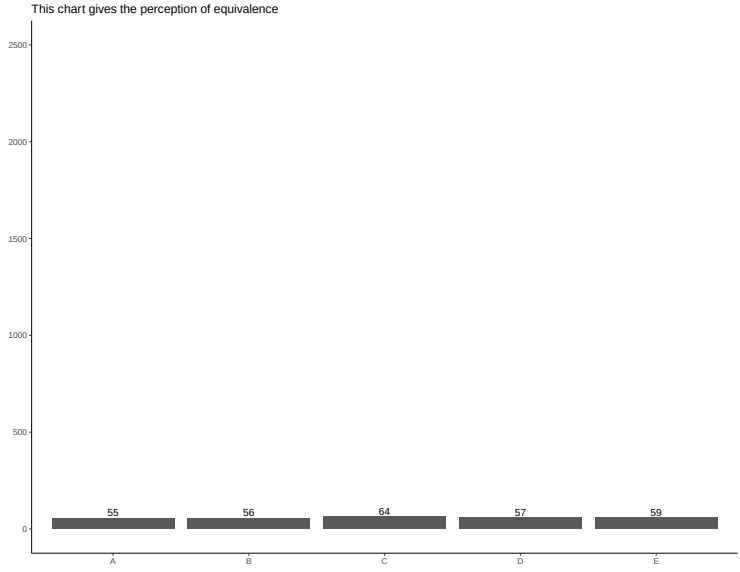
Legibility & Aesthetics

- Design for the "interocular"
- Avoid deceptive or confusing presentation of the data
- Maximize the data-to-ink ratio
- Optimize ticks and gridlines
- Ensure plot is as accessible as possible (e.g., colorblind palette, etc.)
- Simplify wherever possible

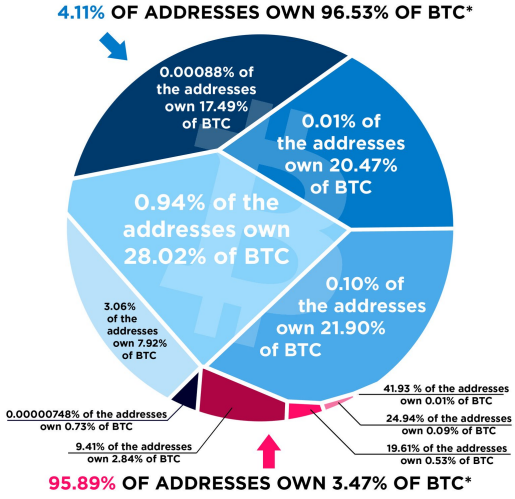
This chart emphasizes the extreme value of C




This chart implies more uniformity
100
75
50
25
0
55
56
64
57
59
A
B
C
D
E


This chart gives the perception of equivalence
2500
2000
1500
1000
500
0
55
56
64
57
59
A
B
C
D
E

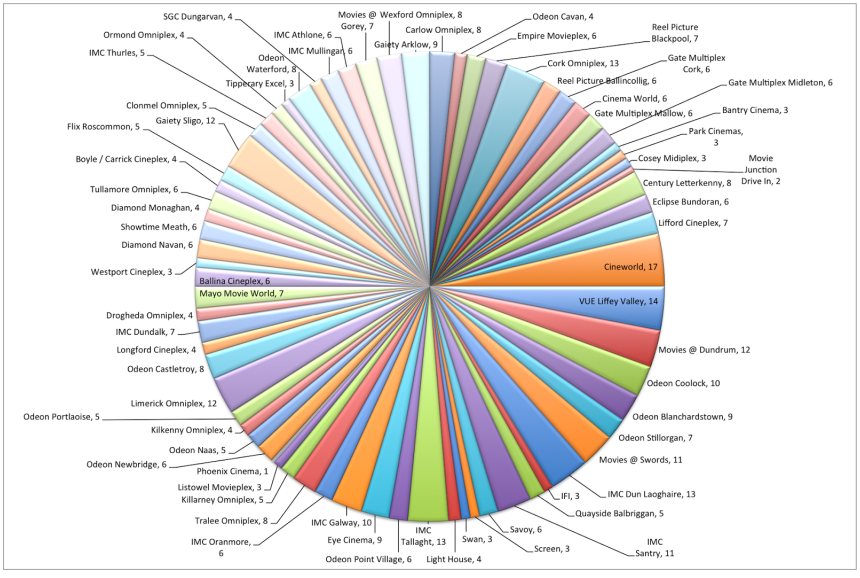
The bitcoin Wealth Distribution



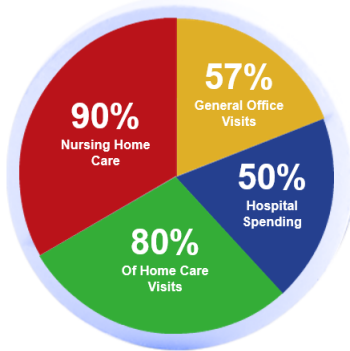
* Data as of September 12th, 2017

Article and Sources:
https://howmuch.net/articles/bitcoin-wealth-distribution
https://bitcoinprivacy.net/

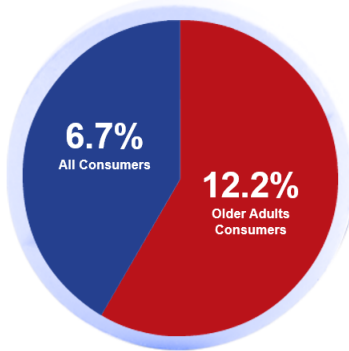
howmuch.net


Movies @ Wexford Omniplex, 8
Odeon Cavan, 4
Carlow Omniplex, 8
Gorey, 7
Empire Movieplex, 6
Gaiety Arklow, 9
Reel Picture Blackpool, 7
SGC Dungarvan, 4
IMC Athlone, 6
Ormond Omniplex, 4
IMC Mullingar, 6
Gate Multiplex Cork, 6
IMC Thurles, 5
Odeon Waterford, 8
Cork Omniplex, 13
Reel Picture Ballincollig, 6
Tipperary Excel, 3
Gate Multiplex Midleton, 6
Cinema World, 6
Clonmel Omniplex, 5
Gate Multiplex Mallow, 6
Bantry Cinema, 3
Gaiety Sligo, 12
Flix Roscommon, 5
Park Cinemas, 3
Cosey Midiplex, 3
Boyle / Carrick Cineplex, 4
Movie Junction Drive In, 2
Century Letterkenny, 8
Tullamore Omniplex, 6
Eclipse Bundoran, 6
Diamond Monaghan, 4
Lifford Cineplex, 7
Showtime Meath, 6
Diamond Navan, 6
Cineworld, 17
Westport Cineplex, 3
Ballina Cineplex, 6
Mayo Movie World, 7
VUE Liffey Valley, 14
Drogheda Omniplex, 4
IMC Dundalk, 7
Movies @ Dundrum, 12
Longford Cineplex, 4
Odeon Castletroy, 8
Odeon Coolock, 10
Limerick Omniplex, 12
Odeon Blanchardstown, 9
Odeon Portlaoise, 5
Kilkenny Omniplex, 4
Odeon Stillorgan, 7
Odeon Naas, 5
Movies @ Swords, 11
Odeon Newbridge, 6
Phoenix Cinema, 1
IMC Dun Laoghaire, 13
Listowel Movieplex, 3
IFI, 3
Killarney Omniplex, 5
Quayside Balbriggan, 5
Tralee Omniplex, 8
Savoy, 6
IMC Galway, 10
IMC Oranmore, 6
Eye Cinema, 9
IMC Tallaght, 13
Swan, 3
Screen, 3
IMC Santry, 11
Odeon Point Village, 6
Light House, 4

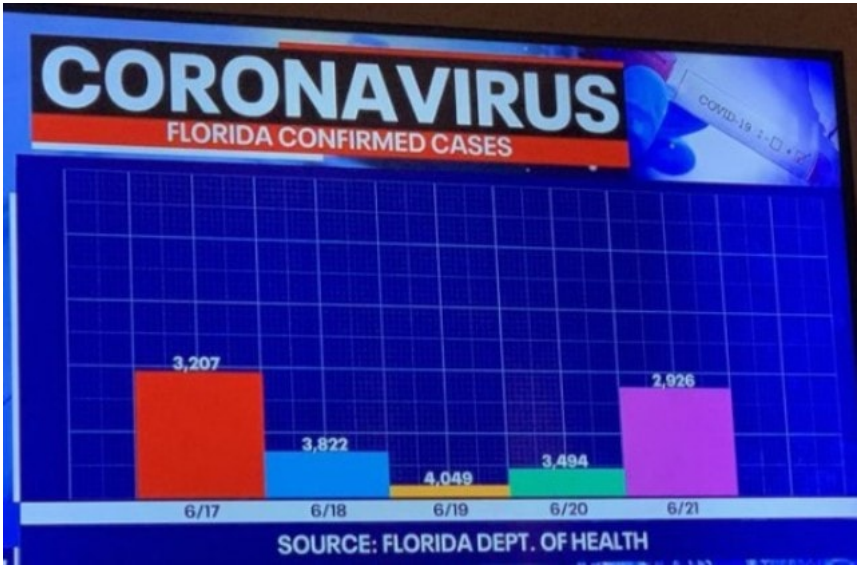
The Numbers on Older Adults

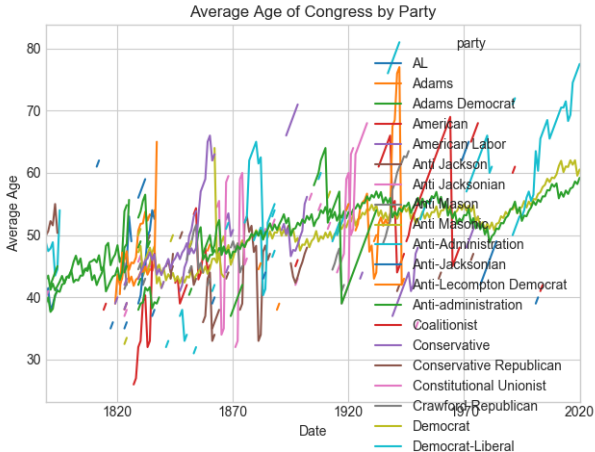


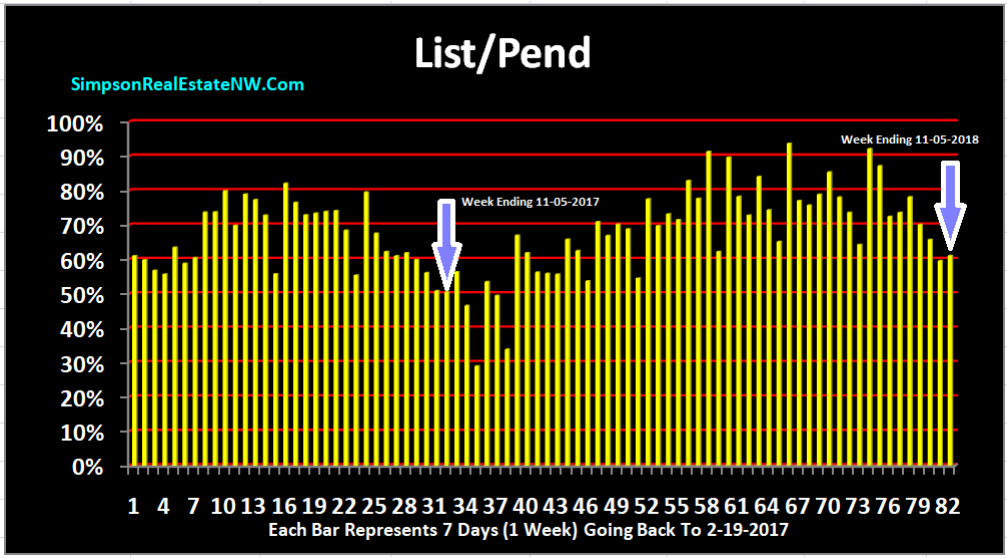
The Majority

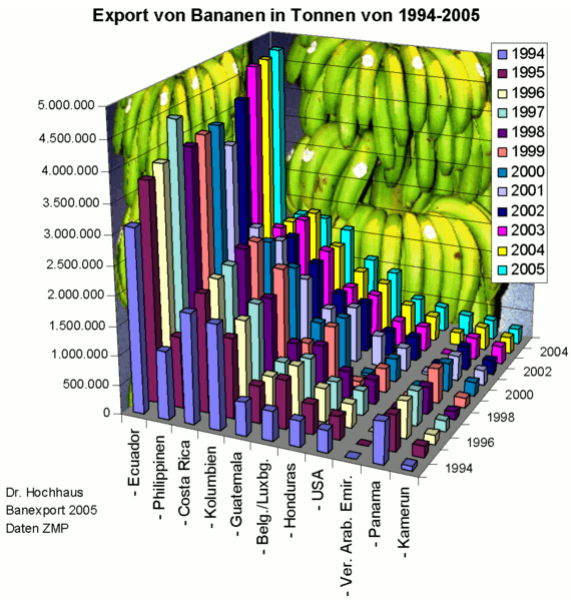


Healthcare Spending


CORONAVIRUS
FLORIDA CONFIRMED CASES
3,207
3,822
4,049
3,494
2,926
6/17
6/18
6/19
6/20
6/21
SOURCE: FLORIDA DEPT. OF HEALTH


Average Age of Congress by Party
Average Age
Date
80
70
60
50
40
30
1820
1870
1920
1970
2020
party
AL
Adams
Adams Democrat
American
American Labor
Anti Jackson
Anti Jacksonian
Anti Mason
Anti Masonic
Anti-Administration
Anti-Jacksonian
Anti-Lecompton Democrat
Anti-administration
Coalitionist
Conservative
Conservative Republican
Constitutional Unionist
Crawford Republican
Democrat
Democrat-Liberal


List/Pend
SimpsonRealEstateNW.Com
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Week Ending 11-05-2017
Week Ending 11-05-2018
1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76 79 82
Each Bar Represents 7 Days (1 Week) Going Back To 2-19-2017


Export von Bananen in Tonnen von 1994-2005
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
5.000.000
4.500.000
4.000.000
3.500.000
3.000.000
2.500.000
2.000.000
1.500.000
1.000.000
500.000
0
- Ecuador
- Philippinen
- Costa Rica
- Kolumbien
- Guatemala
- Belg./Luxbg.
- Honduras
- USA
- Ver. Arab. Emir.
- Panama
- Kamerun
1994
1996
1998
2000
2002
2004
Dr. Hochhaus
Banexport 2005
Daten ZMP

Legibility & Aesthetics

Think through the presentation medium:

- Video and image compression tend to eat at fine details; best to go a bit chonky
- Pure red, green, and blue all tend to bleed, but because of how math works red is most noticeable
- Slides, digital papers, and paper papers all require different font and color selections
- In general, save as .pdf or other vector format (LaTeX handles .pdf well)

Replicability

- Keep a lab notebook
  - Write down **everything**
  - Bookmark useful guides
- Commit to a filing convention and stick to it
- Use a repository such as GitHub for versioning
- Save an archive copy of your data and never touch it again
  - Work from duplicate in your WD
  - Try to avoid overwriting files
- **Everything** gets done in code
- Extensively comment your code
- Save often
  - Make sure autosave is on
  - Back up to the cloud or a secondary drive

Base Graphics

Base Graphics

- Play with some data compiled by IPI–WZB
  Data available on GitHub and at this link: https://wzb-ipi.github.io/corona/
- Includes country-level Covid data and various political correlates

Base Graphics

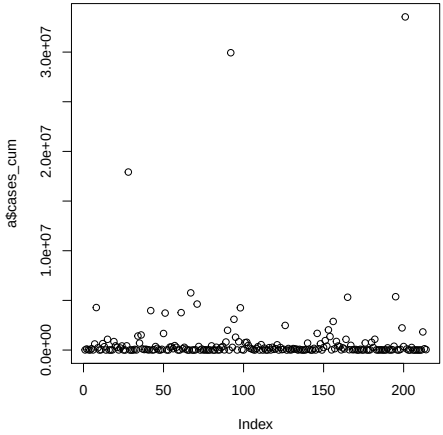
General Format: `plot(x, y, ...)`

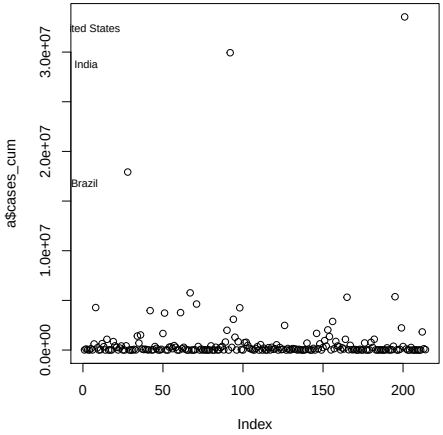
```
model <- lm(y ~ x1, data = df)

pdf("plot.pdf", width = 11, height = 8.5)

par(mfrow = c(1, 2))

plot(df$x, df$y, type = "p")
abline(model, col = "red")

plot(df$x2, df$y,
     xlab = "x label",
     main = "title")

dev.off()
```

Base Graphics

```
a <- read.csv("wzb_covid_june2021.csv", header = TRUE)

# What happens if we just use the command "plot"?
plot(x = a$cases_cum)

# There are some wild outliers.  What countries are these?
plot(x = a$cases_cum)
text(a$cases_cum[a$cases_cum > 15000000],
     labels = a$X[a$cases_cum > 15000000],
     cex = 0.75, pos = 1)
```

Base Graphics

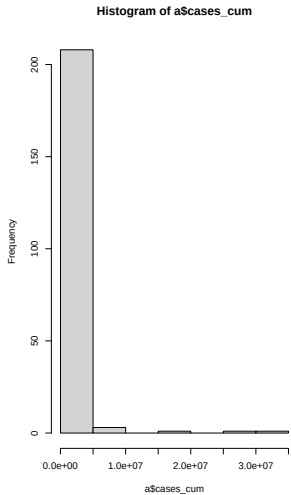


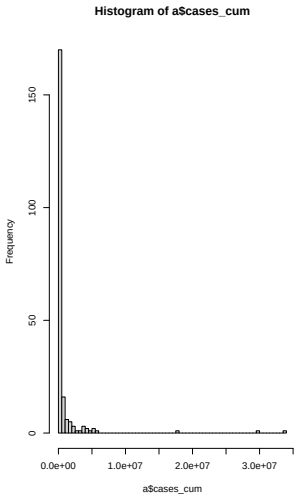


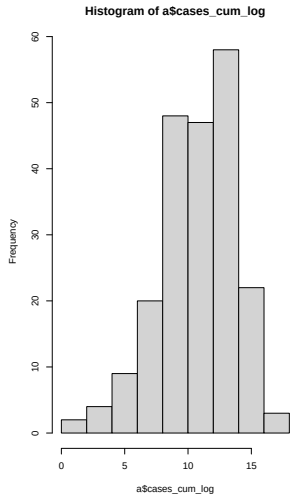
Base Graphics: Histograms

```
# How about a histogram
hist(a$cases_cum)

# Let's add a few more breaks
hist(a$cases_cum, breaks = 50)

# I can't read the exponential notation, so lets try log:
a$cases_cum_log <- log(a$cases_cum)
hist(a$cases_cum_log)
```

Base Graphics: Histogram



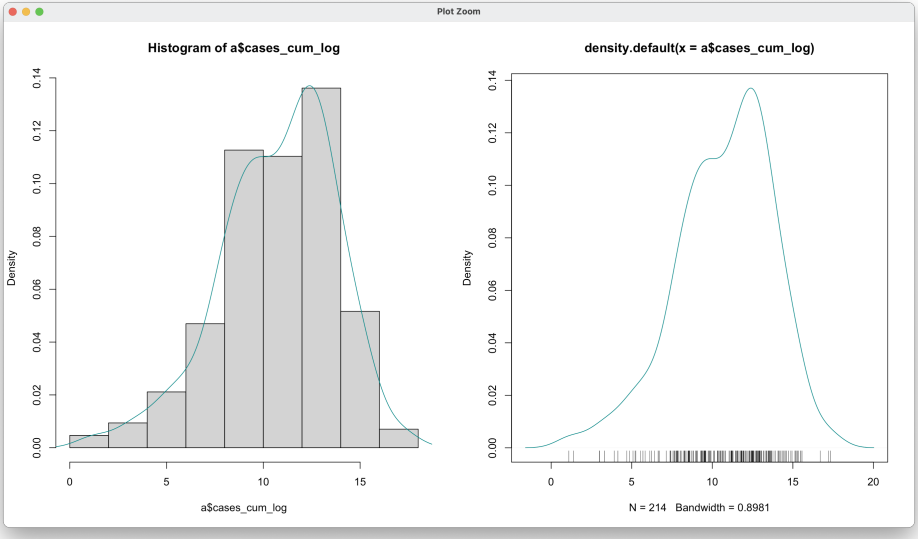




Base Graphics: Histograms, Density, Rugs

```
1  # Let's add the density now
2  d_ccl <- density(a$cases_cum_log)
3
4  # Plot matrix using mfrow / mfcol (rows, columns)
5  par(mfrow = c(1, 2))
6
7  ## Plot 1
8  hist(a$cases_cum_log, freq = FALSE)
9  lines(d_ccl, col = "darkcyan")
10
11 ## Plot 2
12 plot(d_ccl, col = "darkcyan")
13 rug(a$cases_cum_log)
14
15 ## Clear
16 dev.off()
```

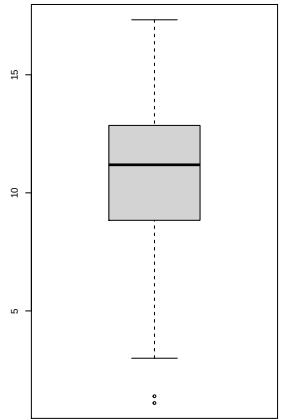
Base Graphics: Histograms, Density, Rugs

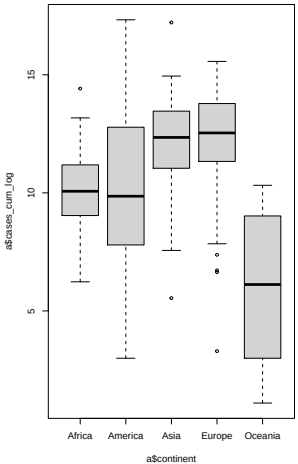


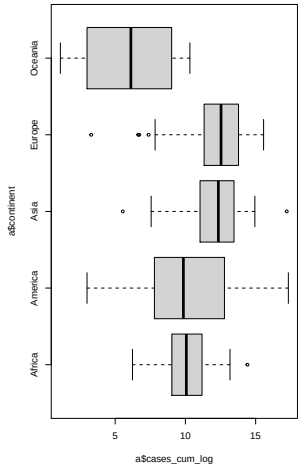
Base Graphics: Box Plots

```
1 boxplot(a$cases_cum_log)
2
3 boxplot(a$cases_cum_log ~ a$continent)
4
5 boxplot(a$cases_cum_log ~ a$continent, horizontal = TRUE)
```

Base Graphics: Box Plots



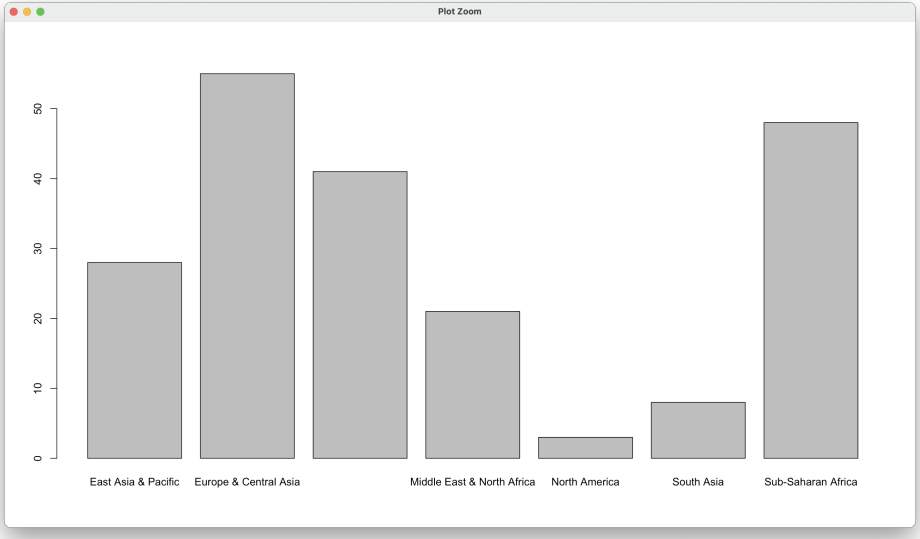




Base Graphics: Bar Plots

```
1 # How many countries are there in each region?
2 b <- table(a$region)
3
4 barplot(b[c(-1)])
```

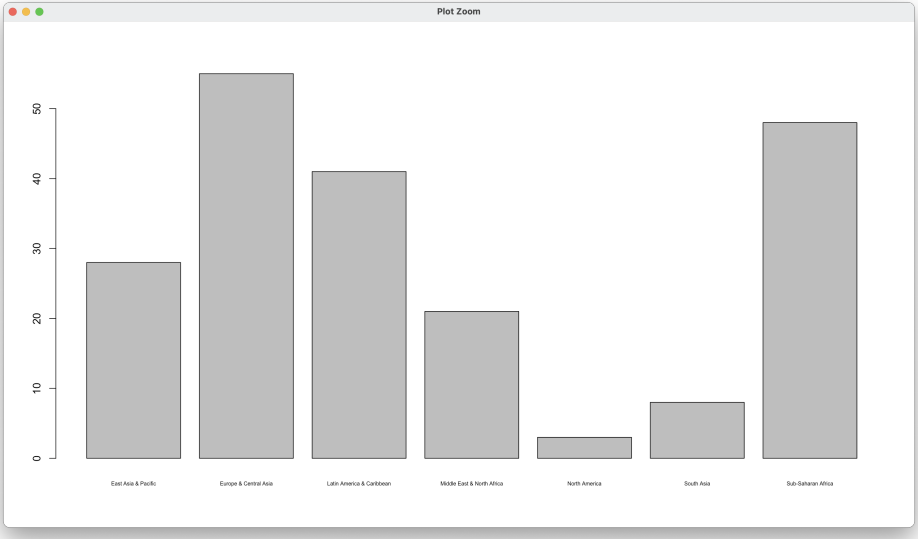
Base Graphics: Bar Plots



Base Graphics: Bar Plots

```
1 # wtf... R isn't plotting all the names?  Because they don't fit...
2 barplot(b[c(-1)], cex.names = 0.5)
```

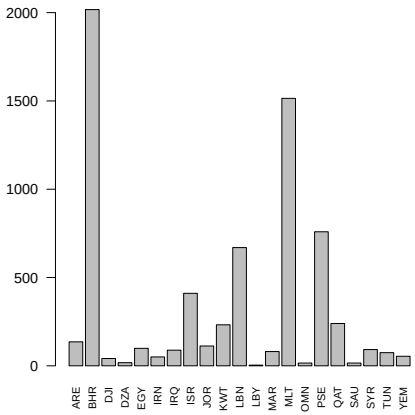
Base Graphics: Bar Plots

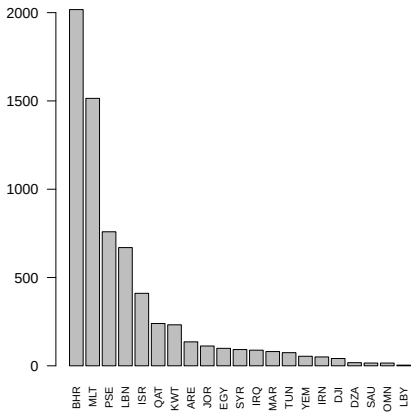


Base Graphics: Bar Plots

```
# Try something else: Filter down to just MENA countries
c <- a %>% filter(region == "Middle East & North Africa")

# New bar plot
barplot(c$pop_density, names.arg = c$geoid2, las = 2, cex.names = 0.75)

# Can we sort these bars?
d <- c[order(c$pop_density, decreasing = TRUE),]
barplot(d$pop_density, names.arg = d$geoid2, las = 2, cex.names = 0.75)
```

Base Graphics: Bar Plots

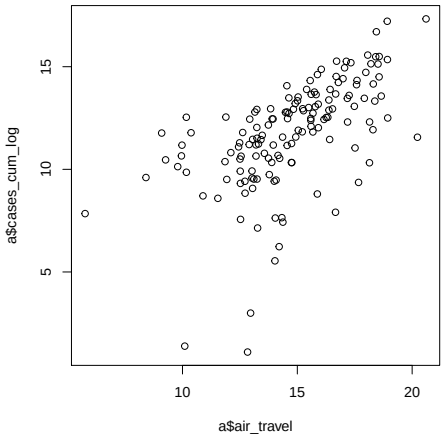


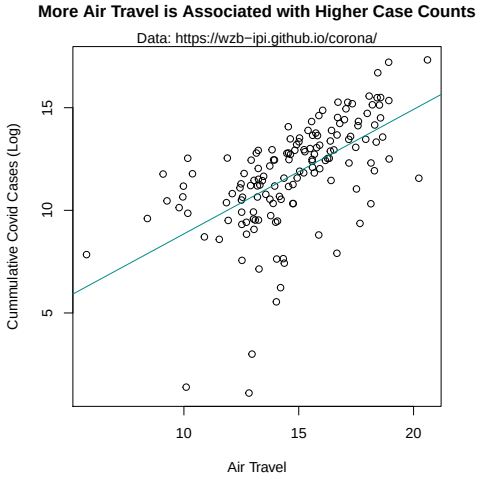


Base Graphics: Scatter Plots

```
plot(a$air_travel, a$cases_cum_log)

# Wow, that was easy.  But let's make it nicer:
plot(a$air_travel, a$cases_cum_log,
     xlab = "Air Travel",
     ylab = "Cummulative Covid Cases (Log)")
abline(lm(a$cases_cum_log ~ a$air_travel), col = "darkcyan")
mtext("Data: https://wzb-ipi.github.io/corona/", side = 3)
title(main = "More Air Travel is Associated with Higher Case Counts")
```

Base Graphics: Scatter Plots

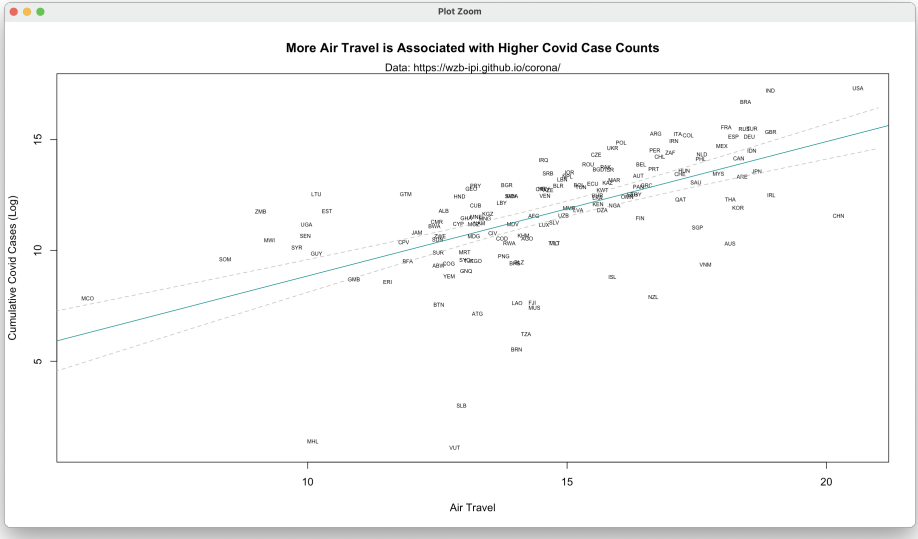




Base Graphics: Scatter Plots

```
# But what about a confidence interval?  We need to run the model and predict:
model <- lm(cases_cum_log ~ air_travel, data = a)
xvalues <- data.frame(air_travel = seq(5 , 21, length.out = 100))
predictions <- predict(model, newdata = xvalues, interval = "confidence")

plot(a$air_travel, a$cases_cum_log,
     col = "white",
     xlab = "Air Travel",
     ylab = "Cumulative Covid Cases (Log)")
text(a$air_travel, a$cases_cum_log, labels = a$geoid2, cex = 0.5)
abline(lm(a$cases_cum_log ~ a$air_travel), col = "darkcyan")
lines(xvalues[,1], predictions[,2], col = "gray", lty = 2)
lines(xvalues[,1], predictions[,3], col = "gray", lty = 2)
mtext("Data: https://wzb-ipi.github.io/corona/", side = 3)
title(main = "More Air Travel is Associated with Higher Covid Case Counts")
```

Base Graphics: Scatter Plots



The Grammar of Graphics

ggplot2

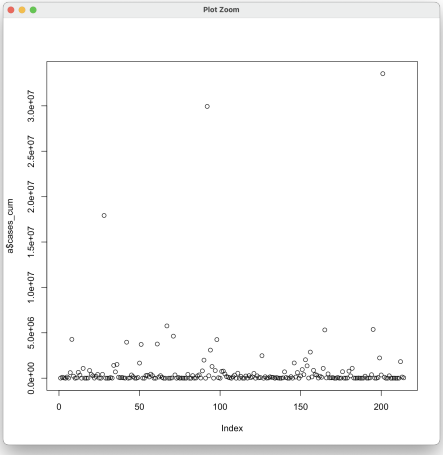
General Format: `ggplot(df, aes(x, y)) + geom_point()`

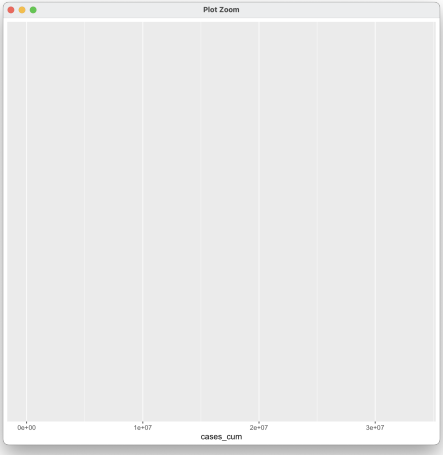
```
1  library(ggplot2)
2
3  plot <- ggplot(df, aes(x = x, y = y)) +
4    geom_point() +
5    geom_smooth(method = "lm") +
6    coord_cartesian(xlim = c(0, 100), ylim = c(0, 100)) +
7    labs(title = "Title",
8         subtitle = "Subtitle",
9         y = "Y Title",
10        x = "X Title",
11        caption = "Caption") +
12   theme_minimal()
13 plot
14 ggsave("plot.pdf", plot = plot, width = 11, height = 8.5)
```

ggplot2

- Remember in base graphics we could generate a plot with:
  `plot(x = a$cases_cum)`
- What happens if we just type:
  `ggplot(a, aes(x = cases_cum))`

ggplot2

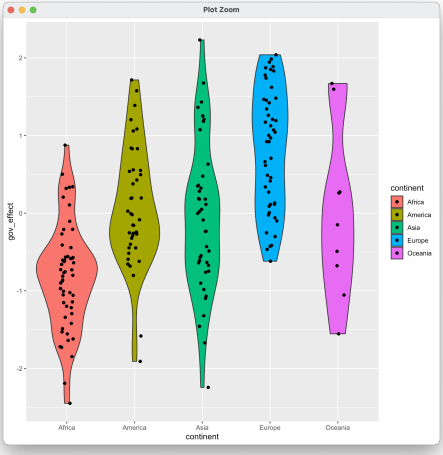

Plot Zoom
aS$cases_cum
0.0e+00
5.0e+06
1.0e+07
1.5e+07
2.0e+07
2.5e+07
3.0e+07
0
50
100
150
200
Index

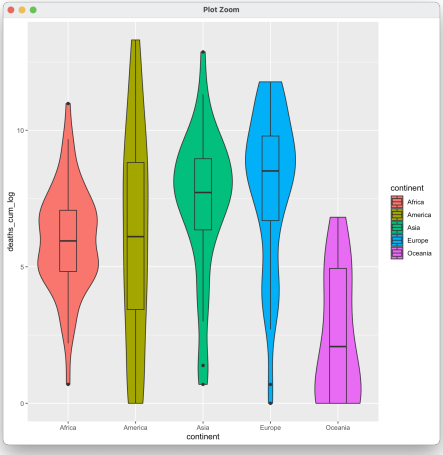

Plot Zoom
0e+00
1e+07
2e+07
3e+07
cases_cum

ggplot2: Violin Plots, Box Plots

```
gg1 <- ggplot(a, aes(x = continent, y = gov_effect, fill = continent)) +
  geom_violin() +
  geom_jitter(position=position_jitter(0.1))
gg1

gg2 <- ggplot(a, aes(x = continent, y = deaths_cum_log, fill = continent)) +
  geom_violin() +
  geom_boxplot(width = 0.25)
gg2
```

ggplot2: Violin Plots, Box Plots

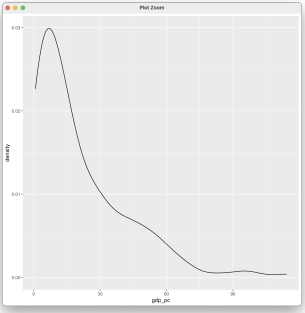


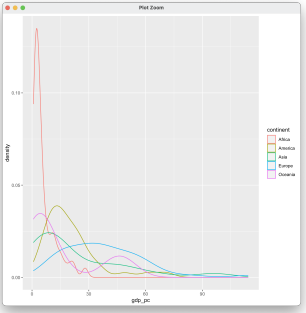


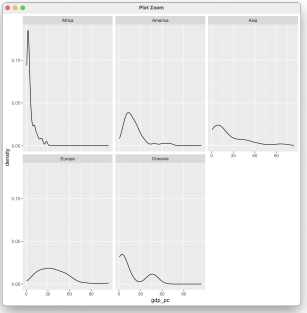
ggplot2: Density Plots

```
gg3 <- ggplot(a, aes(x = gdp_pc)) +
  geom_density()
gg3

gg4 <- ggplot(a, aes(x = gdp_pc)) +
  geom_density(aes(color = continent))
gg4

gg5 <- ggplot(a, aes(x = gdp_pc)) +
  geom_density() +
  facet_wrap(~ continent)
gg5
```

ggplot2:


Plot Zoom
density
gdp_pc

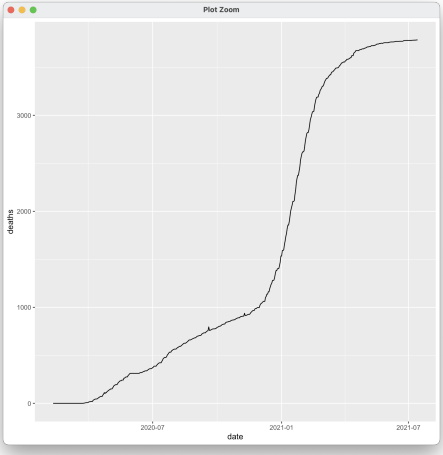

Plot Zoom
density
gdp_pc
continent
Africa
America
Asia
Europe
Oceania

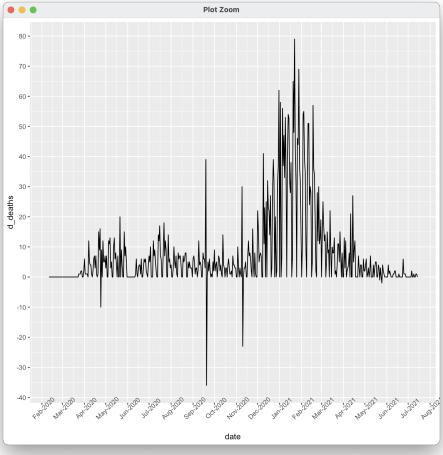

Plot Zoom
Africa
America
Asia
Europe
Oceania
density
gdp_pc

ggplot2: Line / Time Series

```
# Some other Covid data
c  <- covid19(c("US"), level = 3)
d <- c %>% filter(administrative_area_level_3 == "San Diego")

gg6 <- ggplot(d, aes(x = date, y = confirmed)) +
  geom_line()
gg6

gg7 <- ggplot(d) +
  geom_line(aes(x = date, y = deaths))
gg7

# What if we wanted daily deaths?
d <- d %>% mutate(d_deaths = deaths - lag(deaths))

gg8 <- ggplot(d) +
  geom_line(aes(x = date, y = d_deaths)) +
  scale_x_date(date_labels = "%b-%Y", breaks = breaks_pretty(20)) +
  scale_y_continuous(breaks = breaks_pretty(10)) +
  theme(axis.text.x = element_text(angle = 45))
gg8
```

ggplot2: Line / Time Series

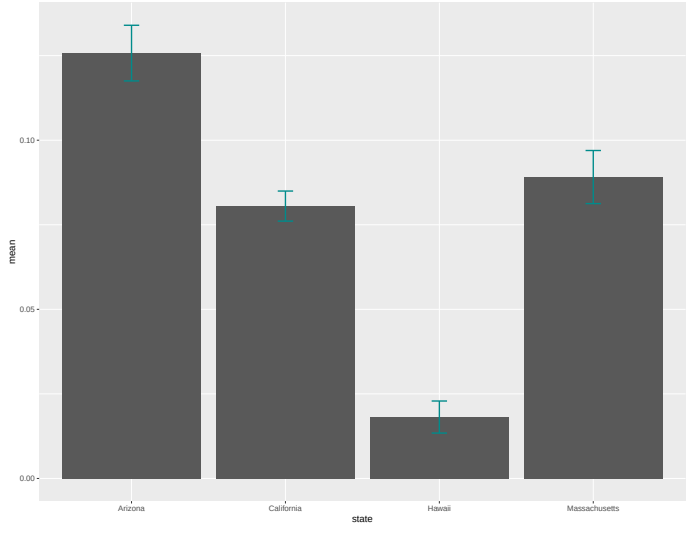




ggplot2: Bar Plots + CI

```r
# Let's look at 1 July in the 4 states that I've lived in for more than a year:
e <- c %>% filter(administrative_area_level_2 == "Hawaii" |
                    administrative_area_level_2 == "California" |
                    administrative_area_level_2 == "Massachusetts" |
                    administrative_area_level_2 == "Arizona")
e <- e %>% filter(date == "2021-07-01")

# Let's normalize the case counts by the population:
h <- e %>% mutate(mconfirmed = confirmed / population) %>%
  data.frame() %>%
  select(state = administrative_area_level_2, mconfirmed) %>%
  group_by(state) %>%
  summarize(n = n(),
            mean = mean(mconfirmed),
            sd = sd(mconfirmed)) %>%
  mutate(se = sd/sqrt(n))

  # Now we can plot:
  gg11 <- ggplot(h) +
  geom_bar(aes(x = state, y = mean), stat = "identity") +
  geom_errorbar(aes(x = state, ymin = mean-se, ymax = mean+se),
                width=0.1, colour="darkcyan", size=0.75)
gg11
```

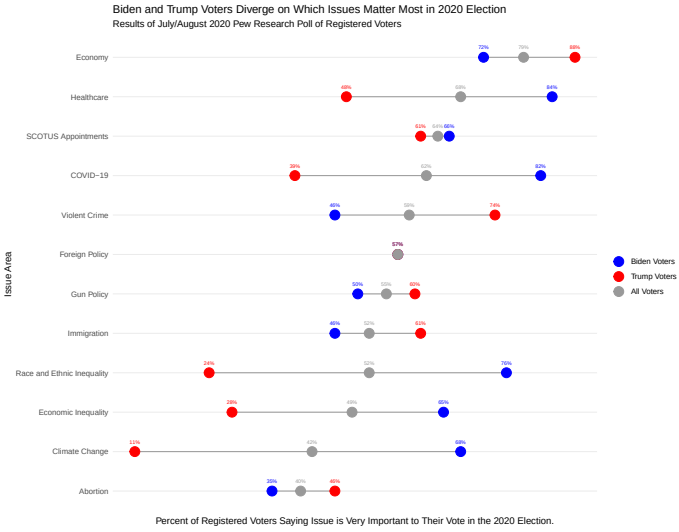
ggplot2: Bar Plots + CI



ggplot2: Dot Plots

```
# Using some Pew data:
b <- read.csv("pew_data.csv")

# We have a problem though! Our data is wide, and we need it to be long...
c <- melt(b, id = "issues")

gg14 <- ggplot(c, aes(x = value, y = reorder(issues, value))) +
  geom_line(aes(group = issues), color = "#999999") +
  geom_point(aes(color = variable), size = 5) +
  scale_color_manual(values = c("#0000FF", "#FF0000", "#999999"),
                     labels = c("Biden Voters", "Trump Voters", "All Voters")) +
  geom_text(aes(label = sprintf("%1.0f%%", value), color = variable),
            size = 2,  nudge_y = 0.25) +
  scale_x_continuous(breaks = NULL) +
  labs(title = "Biden and Trump Voters Diverge on Which Issues Matter Most in 2020 Election",
       subtitle = "Results of July/August 2020 Pew Research Poll of Registered Voters",
       x = "Percent of Registered Voters Saying Issue is Very Important to Their Vote in the 2020 Election.",
       y = "Issue Area") +
  theme_minimal() +
  theme(legend.title = element_blank())
gg14
```

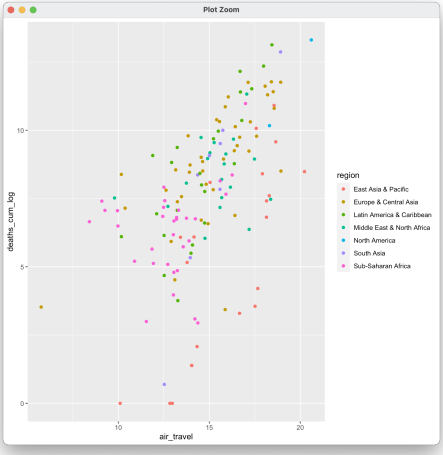
ggplot2: Dot Plots

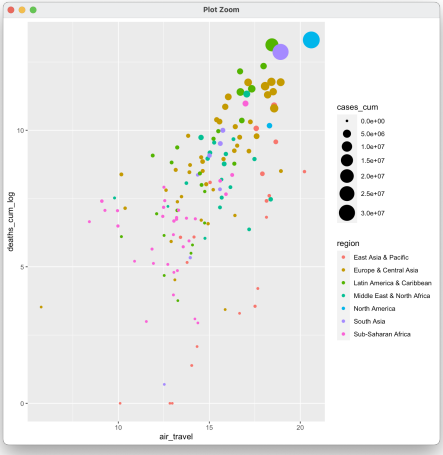


ggplot2: Scatter Plots

```
b <- a %>% filter(region != "")

gg15 <- ggplot(b, aes(x = air_travel, y = deaths_cum_log, color = region)) +
  geom_point()
gg15

# Let's try to scale by case count
gg16 <- ggplot(b, aes(x = air_travel, y = deaths_cum_log,
                      size = cases_cum,
                      color = region)) +
  scale_size_continuous(range = c(1, 10), breaks = pretty_breaks()) +
  geom_point()
gg16
```

ggplot2: Scatter Plots

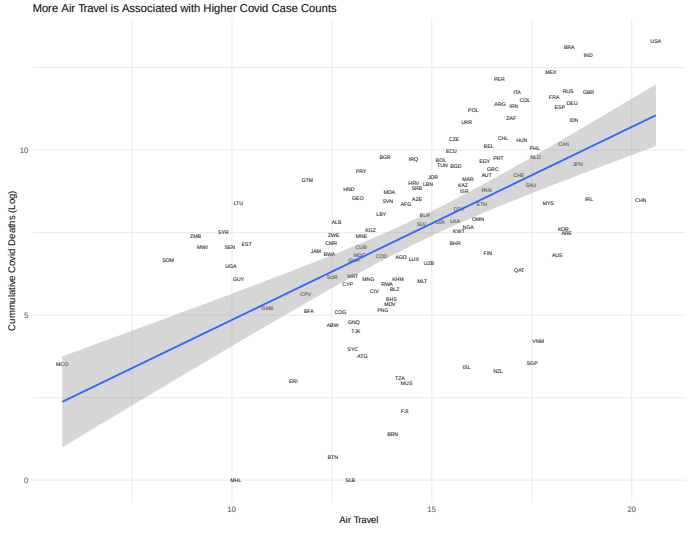




ggplot2: Scatter Plots

```
# Or better yet, can you make it look like the plot we made with base graphics?
gg20 <- ggplot(b, aes(x = air_travel, y = deaths_cum_log)) +
  geom_point() +
  geom_smooth(method = lm) +
  labs(title = "More Air Travel is Associated with Higher Covid Case Counts",
       x = "Air Travel",
       y = "Cummulative Covid Deaths (Log)") +
  theme_minimal()
gg20

# Almost there, but I want ISO codes instead of points
gg20 <- ggplot(b, aes(x = air_travel, y = deaths_cum_log, label = geoid2)) +
  geom_text(size = 2, check_overlap = TRUE) +
  geom_smooth(method = lm,) +
  labs(title = "More Air Travel is Associated with Higher Covid Case Counts",
       x = "Air Travel",
       y = "Cummulative Covid Deaths (Log)") +
  theme_minimal()
gg20
```

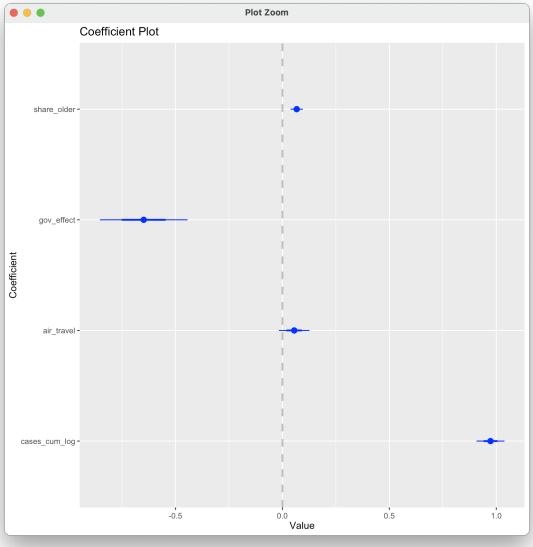
ggplot2: Scatter Plots



ggplot2: Coefficient Plots

```
m <- lm(deaths_cum_log ~ cases_cum_log + air_travel + gov_effect + share_older,
        data = b)
m

coefplot(m, intercept = FALSE)
# Wow, that was easy
```

ggplot2: Coefficient Plots



ggplot2: Coefficient Plots

Wrangle the data to make a coefficient plot by hand:

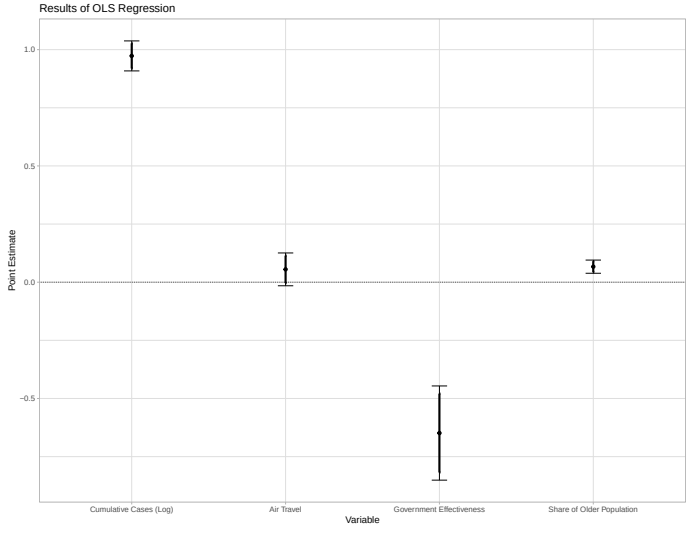
```
# Can we make it harder?  Let's try to replicate this manually in ggplot
z1 <- tidy(m, conf.int = TRUE, conf.level = .95)
z1 <- z1 %>% dplyr::rename(low95 = conf.low, high95 = conf.high)

z2 <- tidy(m, conf.int = TRUE, conf.level = .9)
z2 <- z2 %>% dplyr::rename(low90 = conf.low, high90 = conf.high)

z3 <- z1 %>% full_join(z2)
z3$term <- dplyr::recode(z3$term,
                    "(Intercept)" = "Intercept",
                    "cases_cum_log" = "Cumulative Cases (Log)",
                    "air_travel" = "Air Travel",
                    "gov_effect" = "Government Effectiveness",
                    "share_older" = "Share of Older Population")

z3$term <- factor(z3$term,
                  levels = c("Intercept",
                             "Cumulative Cases (Log)",
                             "Air Travel",
                             "Government Effectiveness",
                             "Share of Older Population"),
                  ordered = TRUE)
```

ggplot2: Coefficient Plots

Now we can actually plot it:

```
gg21 <- ggplot(z3 %>% filter(term != "Intercept")) +
  geom_hline(yintercept=0, lty="11", colour="grey30") +
  geom_errorbar(aes(term, ymin = low95, ymax = high95), width=0.1) +
  geom_errorbar(aes(term, ymin = low90, ymax = high90), lwd = 1.15, width=0) +
  geom_point(aes(term, estimate)) +
  labs(title = "Results of OLS Regression",
       y = "Point Estimate",
       x = "Variable") +
  theme_light()
gg21
```

ggplot2: Coefficient Plots



Final Tips

- Sketch out your plot on paper before you begin
- Build plots iteratively, one element at a time
  - ggplot is particularly conducive to this method
  - Allows you to see what works and what doesn't
- Try to write generic and reusable code
- Remember that coding is nothing more than learning what to ask Google so that it gives you a useful response
  - You are almost certainly not the first person to ever encounter a particular coding issue
  - Don't forget about YouTube; if it's not on YouTube it doesn't exist
- Remember that R isn't the only statistical programming package out there

Feel Free to Reach Out!

Mike Seese

mseese@ucsd.edu

@mikeseese

https://github.com/mfseese