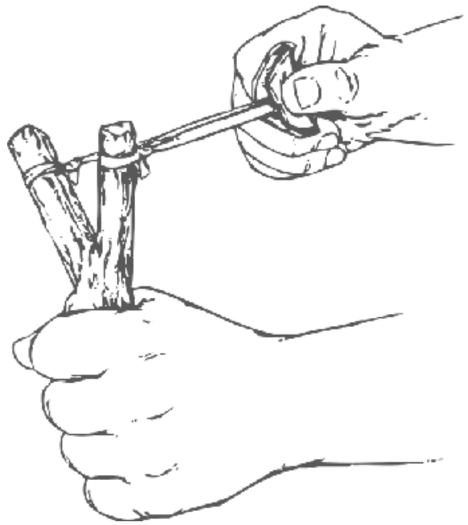
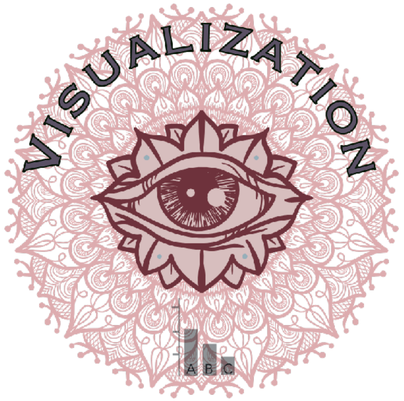
INTRODUCTION TO DATA ANALYSIS

DATA VISUALIZATION

LEARNING GOALS



- obtain a basic understanding of better/worse plotting
  - understand the idea of hypothesis-driven visualization
- develop a basic understanding of the 'grammar of graphs'
- get familiar with frequent visualization strategies
  - barplots, densities, violins, error bars etc.
- be able to fine-tune graphs for better visualization



WHY VISUALIZE?

- a picture can be worth a million words (and numbers)
- every data analysis should start with a ‘getting to know the data’ phase
  - visualization of different aspects of data is key to get intimate with the data
- data visualization as a means of communication (with others)
  - hypothesis-driven visualization: obtain visual (suggestive) evidence regarding a research question of relevance

WHY VISUALIZE?

- a picture can be worth a million words (and numbers)
  - summary statistics can be misleading (because of information loss)
- every data analysis should start with a ‘getting to know the data’ phase
  - use extensive visualization to get intimate with the data
- data visualization as a means of communication (with others / with yourself)
  - hypothesis-driven visualization: obtain visual (suggestive) evidence regarding a research question of relevance

BEYOND SUMMARY STATISTICS

MOTIVATING EXAMPLE :: ANSCOMBE'S QUARTET

- famous data set, ships with core R

```
glimpse(anscombe %>% as_tibble)
```

```
## Observations: 11
## Variables: 8
## $ x1 <dbl> 10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5
## $ x2 <dbl> 10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5
## $ x3 <dbl> 10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5
## $ x4 <dbl> 8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8
## $ y1 <dbl> 8.04, 6.95, 7.58, 8.81, 8.33, 9.96,
## $ y2 <dbl> 9.14, 8.14, 8.74, 8.77, 9.26, 8.10,
## $ y3 <dbl> 7.46, 6.77, 12.74, 7.11, 7.81, 8.84,
## $ y4 <dbl> 6.58, 5.76, 7.71, 8.84, 8.47, 7.04,
```

messy start

```
tidy_anscombe <- anscombe %>% as_tibble %>%
  pivot_longer(
    ## we want o pivot every column
    everything(),
    ## use reg-exps to capture 1st and 2nd character
    names_pattern = "(.)(.)",
    ## assign names to new cols, using 1st part of
    ##  what reg-exp captures as new column names
    names_to = c(".value", "grp")
  ) %>%
  mutate(grp = paste0("Group ", grp))
tidy_anscombe
```

tidy up

```
## # A tibble: 44 x 3
##    grp         x     y
##    <chr>   <dbl> <dbl>
##  1 Group 1    10  8.04
##  2 Group 2    10  9.14
##  3 Group 3    10  7.46
##  4 Group 4     8  6.58
##  5 Group 1     8  6.95
##  6 Group 2     8  8.14
##  7 Group 3     8  6.77
##  8 Group 4     8  5.76
##  9 Group 1    13  7.58
## 10 Group 2    13  8.74
## # ... with 34 more rows
```

nice!

MOTIVATING EXAMPLE :: ANSCOMBE'S QUARTET

```
## # A tibble: 44 x 3
##    grp          x     y
##    <chr>    <dbl> <dbl>
##  1 Group 1     10  8.04
##  2 Group 2     10  9.14
##  3 Group 3     10  7.46
##  4 Group 4      8  6.58
##  5 Group 1      8  6.95
##  6 Group 2      8  8.14
##  7 Group 3      8  6.77
##  8 Group 4      8  5.76
##  9 Group 1     13  7.58
## 10 Group 2     13  8.74
## # ... with 34 more rows
```

input data

```
tidy_anscombe %>%
  group_by(grp) %>%
  summarise(
    mean_x    = mean(x),
    mean_y    = mean(y),
    min_x     = min(x),
    min_y     = min(y),
    max_x     = max(x),
    max_y     = max(y),
    crrltn    = cor(x,y)
  )
```

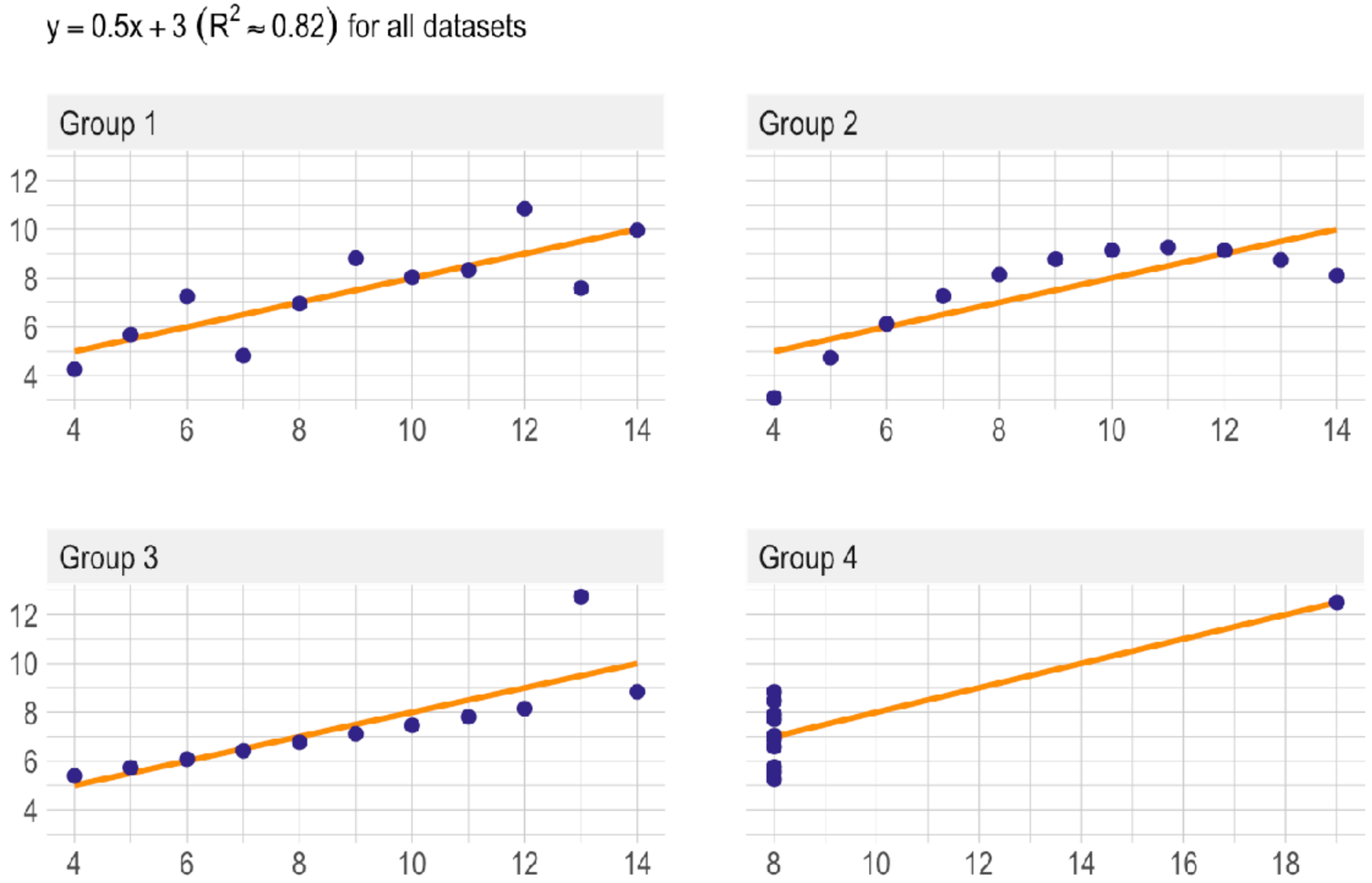
summarise

```
## # A tibble: 4 x 8
##   grp     mean_x mean_y min_x min_y max_x max_y crrltn
##   <chr>    <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl>
## 1 Group 1      9   7.50     4  4.26    14 10.8   0.816
## 2 Group 2      9   7.50     4  3.1     14  9.26  0.816
## 3 Group 3      9   7.5      4  5.39    14 12.7   0.816
## 4 Group 4      9   7.50     8  5.25    19 12.5   0.817
```

all four groups look very similar!

MOTIVATING EXAMPLE :: ANSCOMBE'S QUARTET

- quite different patterns despite similar correlation



PRINCIPLES OF GOOD VISUALIZATION

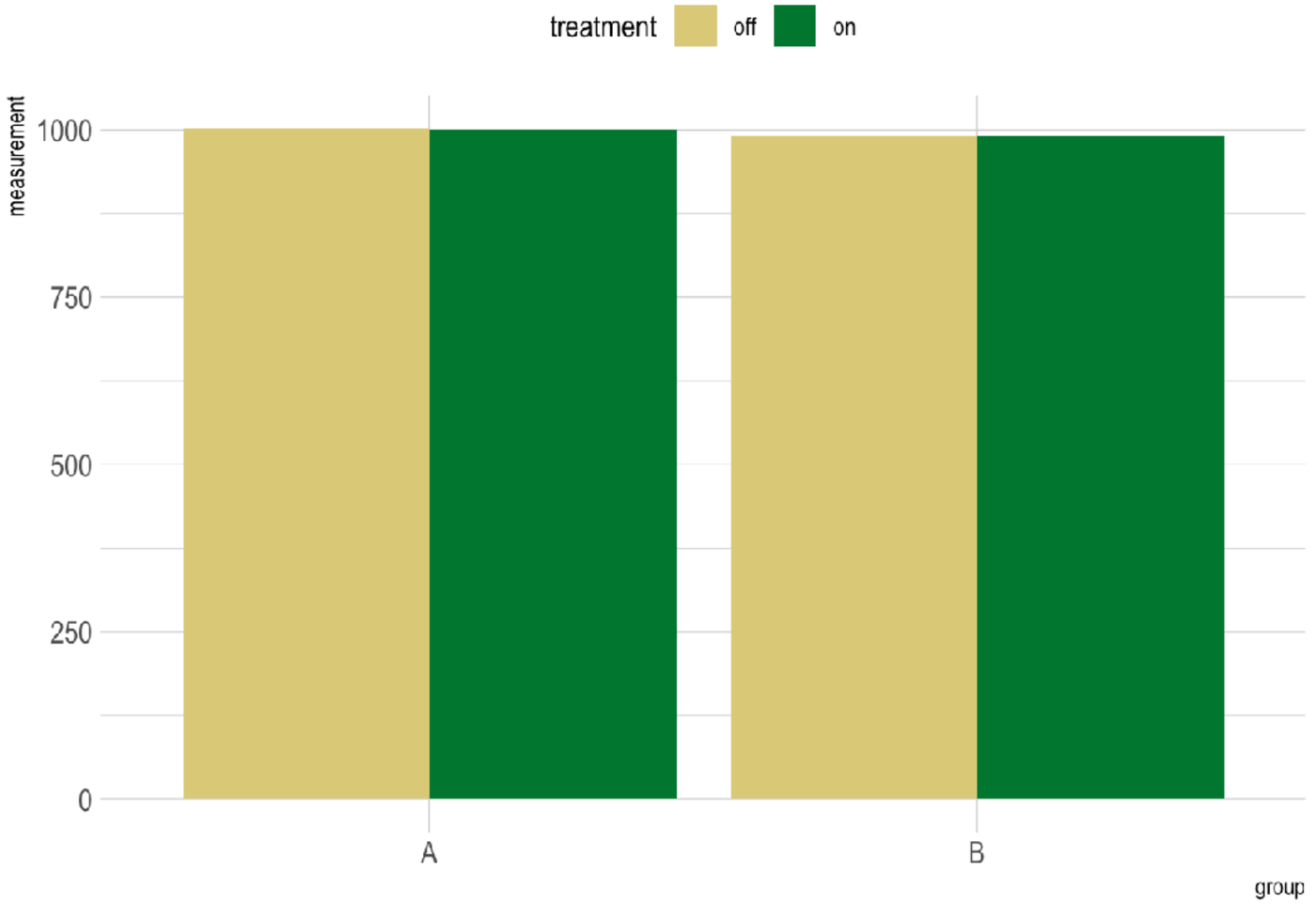
- maximize data-ink ratio (Tufte 1983)
  - maximize information, minimize ink
  - contra chart junk
  - ink vs. processing effort
- analogy to language
  - information flow
  - ease of processing
  - bound by conventional rules
- hypothesis-driven visualization
  - relevance of information

The vague & defeasible rule of thump of good data visualization (according to the author).

"Communicate a maximal degree of relevant true information in a way that minimizes the recipient's effort of retrieving this information."

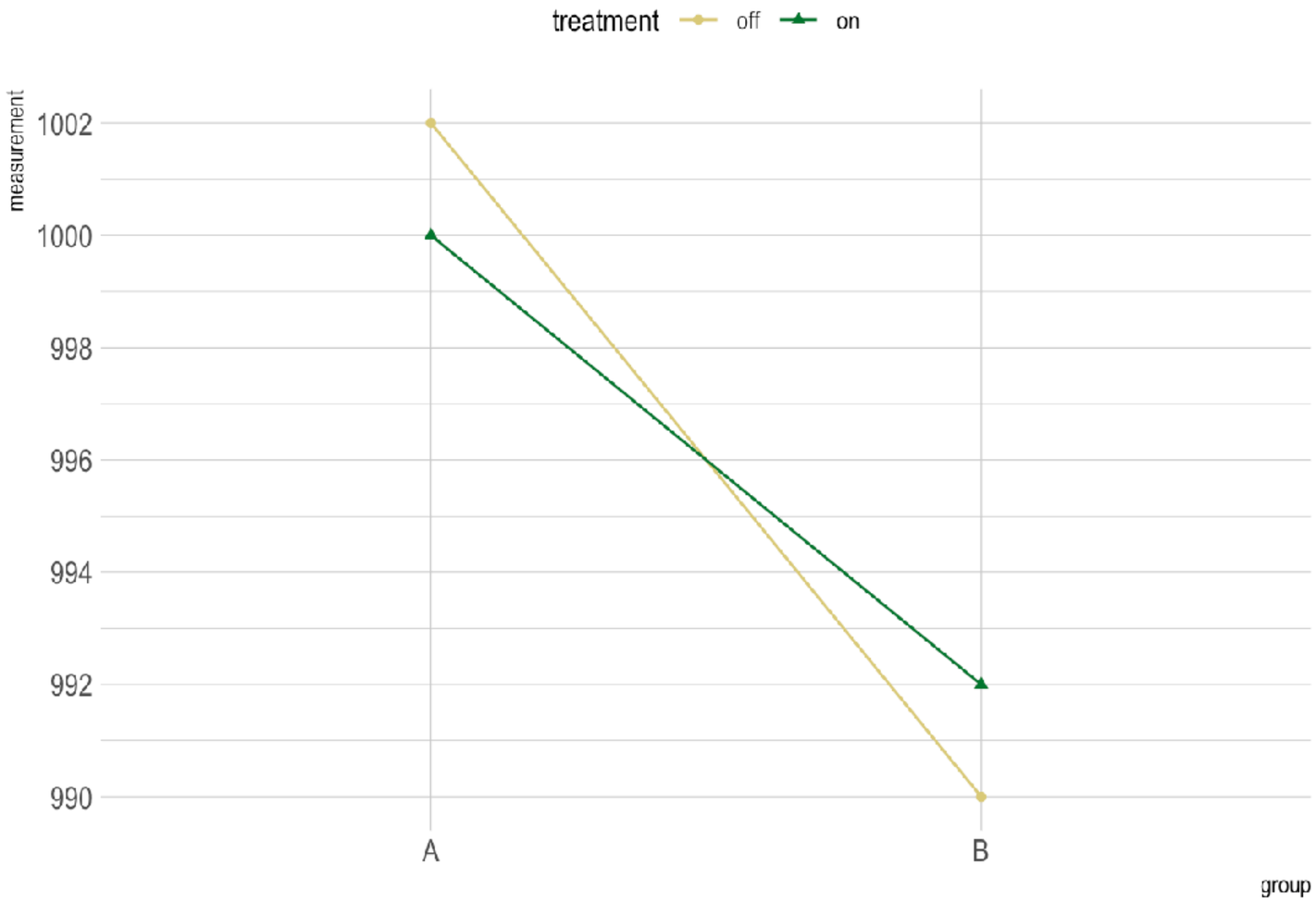
EXAMPLE OF UNINFORMATIVE PLOTTING

```
large_contrast_data <- tribble(
  ~group, ~treatment, ~measurement,
  "A",    "on",       1000,
  "A",    "off",      1002,
  "B",    "on",       992,
  "B",    "off",      990
)
```



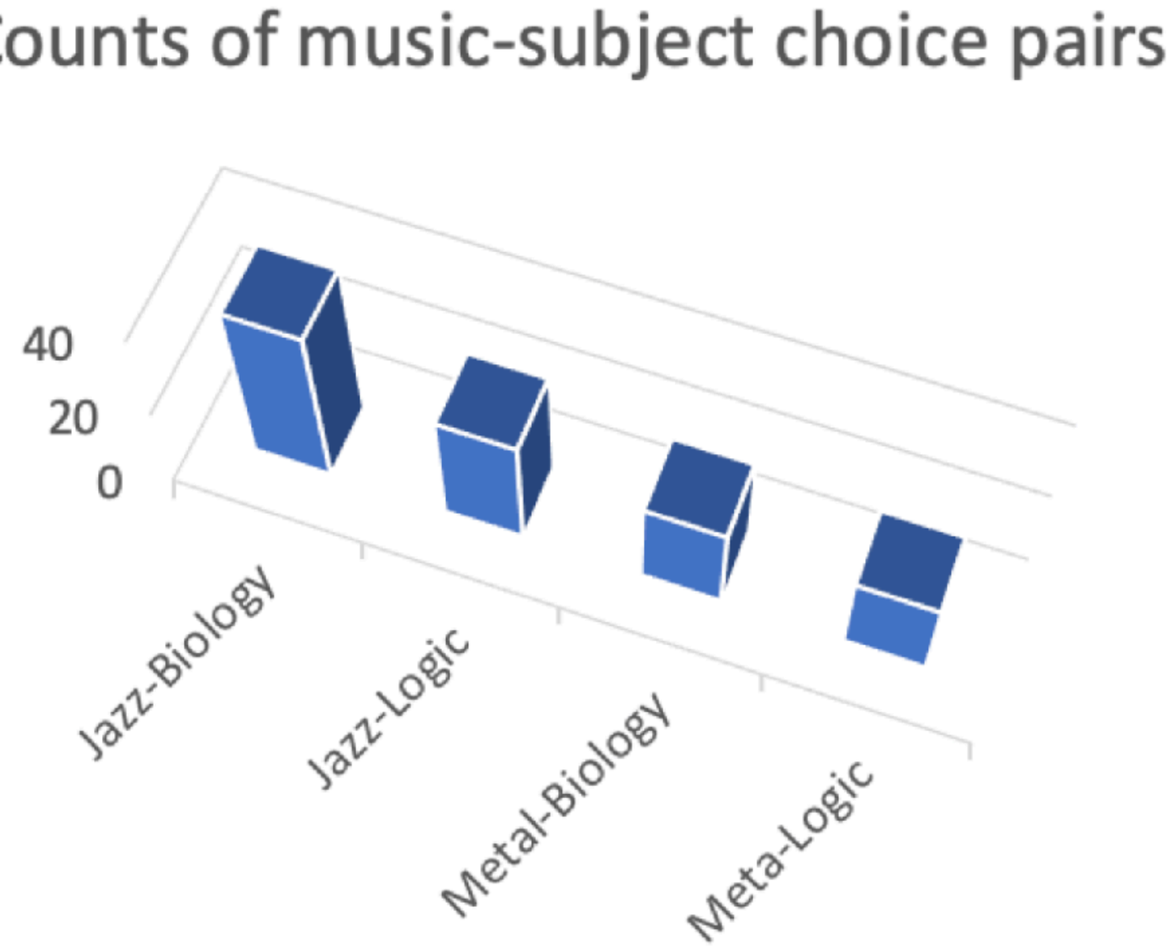
EXAMPLE OF INFORMATIVE HYPOTHESIS-DRIVEN PLOTTING

```
large_contrast_data <- tribble(
  ~group, ~treatment, ~measurement,
  "A",    "on",       1000,
  "A",    "off",      1002,
  "B",    "on",       992,
  "B",    "off",      990
)
```



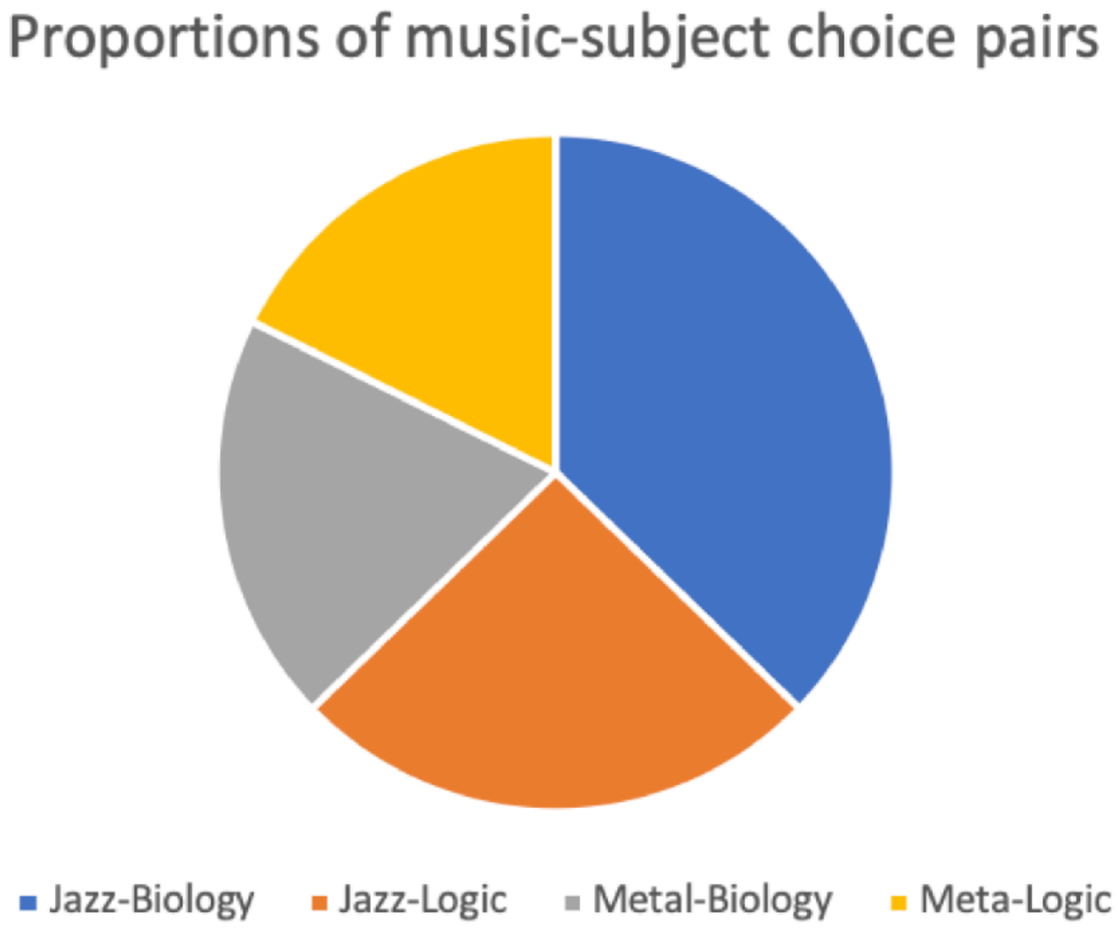
EXAMPLE OF UNINFORMATIVE PLOTTING

```
## # A tibble: 4 x 3
##   JM    LB          n
##   <chr> <chr>   <int>
## 1 Jazz  Biology    38
## 2 Jazz  Logic      26
## 3 Metal Biology    20
## 4 Metal Logic      18
```



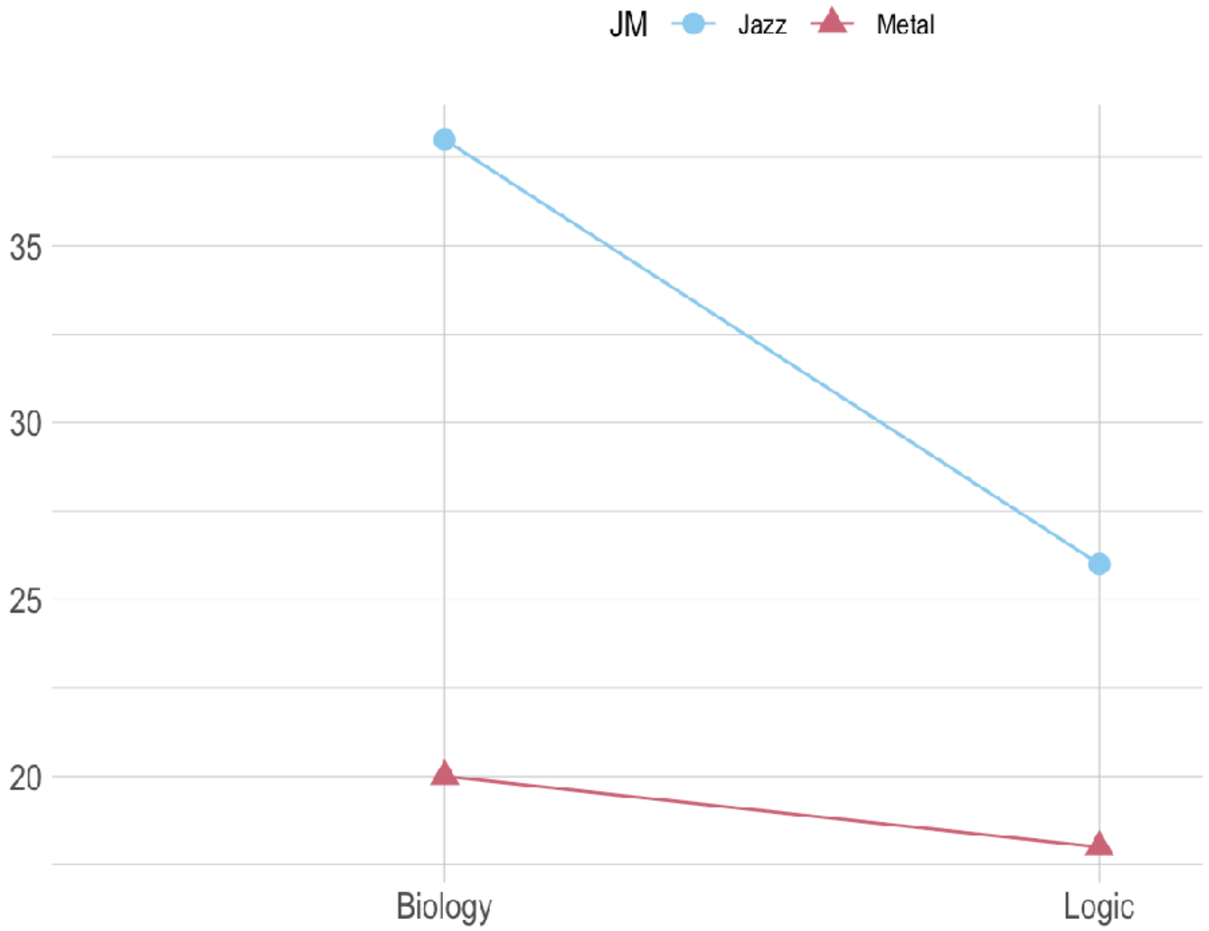
EXAMPLE OF (STILL) UNINFORMATIVE PLOTTING

```
## # A tibble: 4 x 3
##   JM    LB          n
##   <chr> <chr>   <int>
## 1 Jazz  Biology    38
## 2 Jazz  Logic      26
## 3 Metal Biology    20
## 4 Metal Logic      18
```



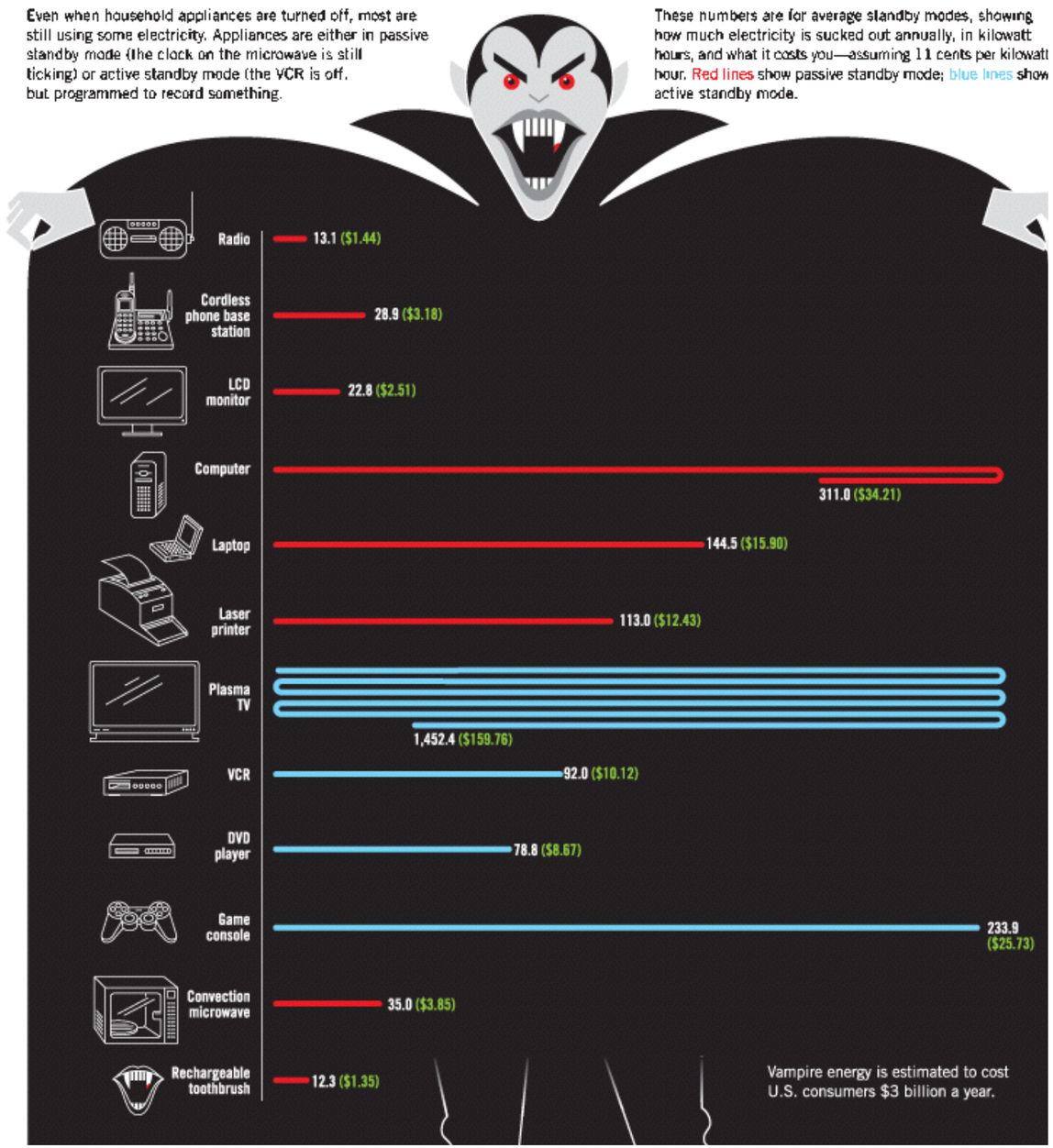
EXAMPLE OF INFORMATIVE HYPOTHESIS-DRIVEN PLOTTING

```
## # A tibble: 4 x 3
##   JM    LB          n
##   <chr> <chr>   <int>
## 1 Jazz  Biology    38
## 2 Jazz  Logic      26
## 3 Metal Biology    20
## 4 Metal Logic      18
```



INFOGRAPHICS

- ≠ hypothesis-driven visualization
- purposes:
  - memorability
  - eye-catchiness
  - persuasion
  - ....



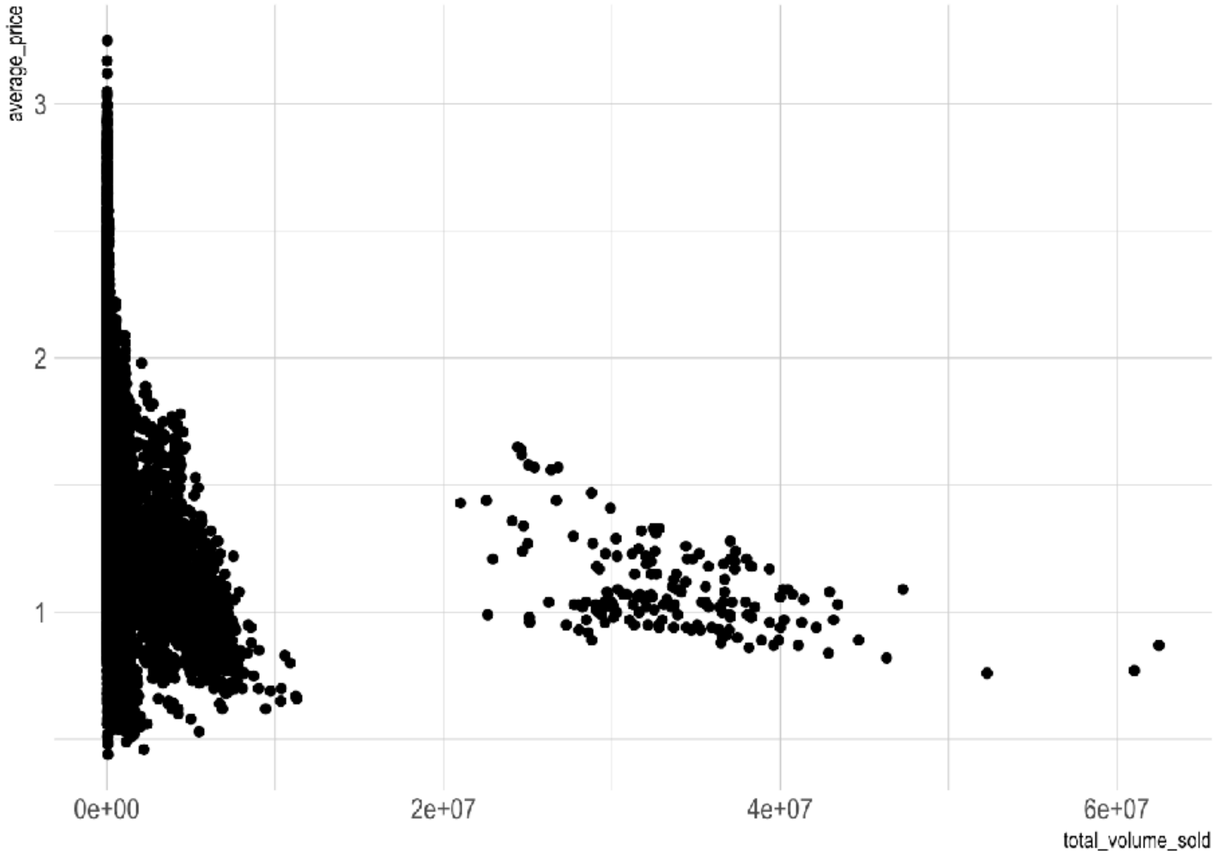
LAYERING AND DEFAULTS

data -> transformation -> geom. object -> aesthetics

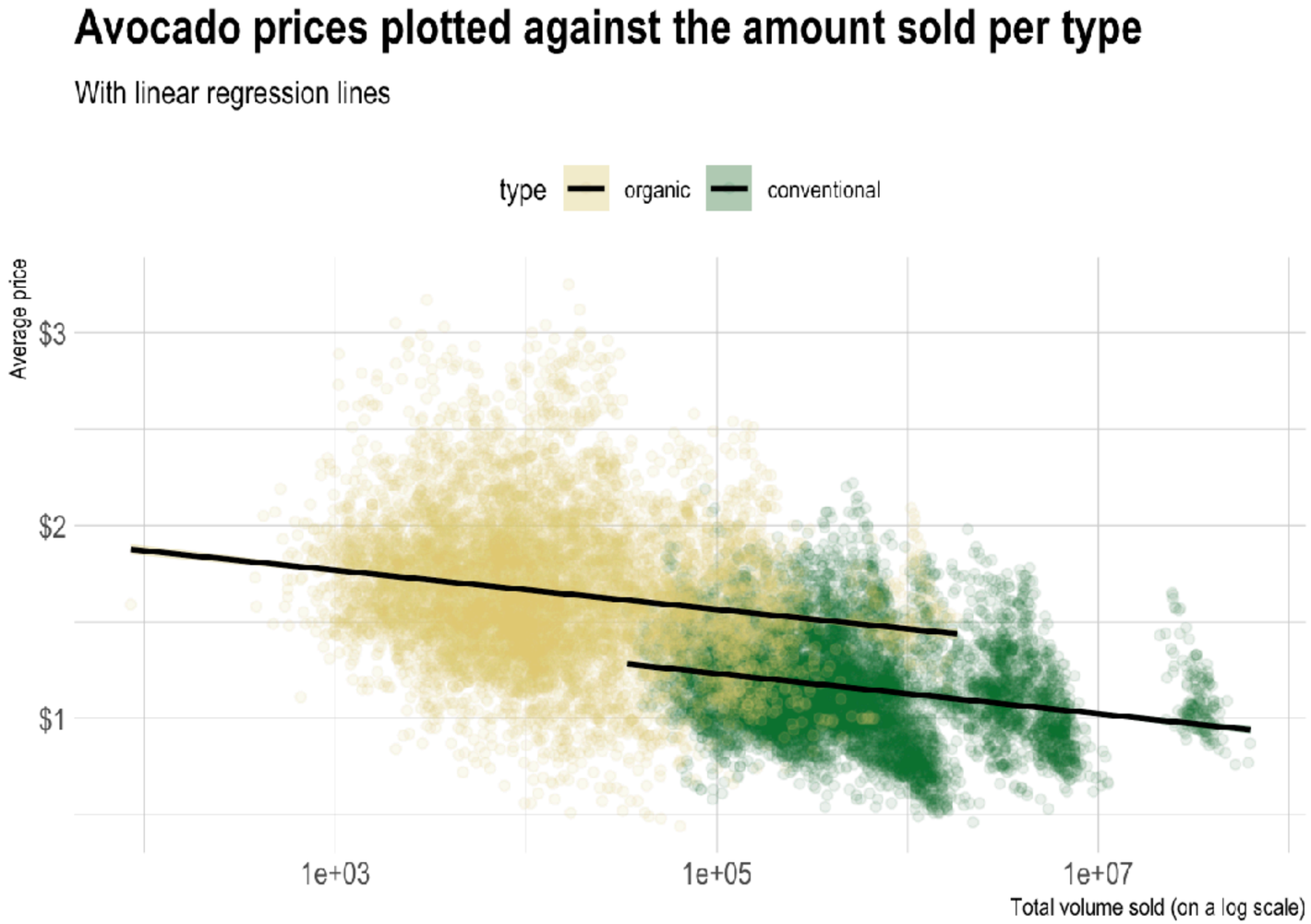
- equivalent to the code before

explicit calls

```
ggplot() +
  layer(
    data = avocado_data,
    mapping = aes(x = total_volume_sold, y = average_price),
    geom = "point",
    stat = "identity",
    position = "identity"
  ) +
  scale_x_continuous() +
  scale_y_continuous() +
  coord_cartesian()
```



CUSTOMIZING A PLOT



CUSTOMIZING A PLOT

```
# pipe data set into function `ggplot`
avocado_data %>%
  # reverse factor level so that horizontal legend entries align with
  # the majority of observations of each group in the plot
  mutate(
    type = fct_rev(type)
  ) %>%
  # initivalize the plot
  ggplot(
    # defined mapping
    mapping = aes(
      # which variable goes on the x-axis
      x = total_volume_sold,
      # which variable goes on the y-axis
      y = average_price,
      # which groups of variables to distinguish
      group = type,
      # color and fill to change by grouping variable
      fill = type,
      color = type
    )
  ) +
```

```
  # declare that we want a scatter plot
  geom_point(
    # set low opacity for each point
    alpha = 0.1
  ) +
  # add a linear model fit (for each group)
  geom_smooth(
    color = "black",
    method = "lm"
  ) +
  # change the default (normal) of x-axis to log-scale
  scale_x_log10() +
  # add dollar signs to y-axis labels
  scale_y_continuous(labels = scales::dollar) +
  # change axis labels and plot title & subtitle
  labs(
    x = 'Total volume sold (on a log scale)',
    y = 'Average price',
    title = "Avocado prices plotted against the amount sold per type",
    subtitle = "With linear regression lines"
  )
```