432 Class 2 Slides

github.com/THOMASELOVE/2020-432

2020-01-16

from *The Art of Statistics*

Chapter 2: Summarizing and Communicating Numbers. Lots of Numbers.

- A variety of statistics can be used to summarize the empirical distribution of data points, including measures of location and spread.
- Skewed data distributions are common, and some summary statistics are very sensitive to outlying values.
- Data summaries always hide some detail, and care is required so that important information is not lost.
- Single sets of numbers can be visualised in strip-charts, box-and-whisker plots and histograms.
- Consider transformations to better reveal patterns, and use the eye to detect patterns, outliers, similarities and clusters.

(*list continues on next slide*)

The Art of Statistics

Chapter 2: Summarizing and Communicating Numbers. Lots of Numbers.

(*continuing from previous slide*)

- Look at pairs of numbers as scatter-plots, and time series as line-graphs.
- When exploring data, a primary aim is to find factors that explain the overall variation.
- Graphics can be both interactive and animated.
- Infographics highlight interesting features and can guide the viewer through a story, but should be used with awareness of their purpose and their impact.

How might we mostly effectively summarize these data?

Question 1. Excitement about statistics and data science?

- 1 = I have nightmares about this class.
- 10 = Nate Silver is my hero.

45566 77777 77788 88888 88888 88999 99999 99999 99000 00000

Question 2. Interest in US Democratic Primary?

- 10 = I am obsessed with it.
- 1 = I would have difficulty caring less.

00009 99999 98888 88888 88877 77777 76666 65555 55422 22211

Working with a Large Survey

BRFSS and SMART

The Centers for Disease Control analyzes Behavioral Risk Factor Surveillance System (BRFSS) survey data for specific metropolitan and micropolitan statistical areas (MMSAs) in a program called the Selected Metropolitan/Micropolitan Area Risk Trends of BRFSS (SMART BRFSS.)

In this work, we will focus on data from the 2017 SMART, and in particular on data from the Cleveland-Elyria, OH, Metropolitan Statistical Area.

Note that the Course Notes (from Chapter 2) describe the work of cleaning the data in gruesome detail. Today, we'll work with a smaller chunk of the data developed there.

Setup

```
library(here); library(magrittr); library(janitor)
library(broom); library(simputation); library(patchwork)
library(tidyverse)

theme_set(theme_bw())

smart0 <- read_csv(here("data/smart_ohio.csv"))
```

Get the data on the Data and Code page (green button to download all)

Winnowing the Variables

```r
dim(smart0)
```

```
[1] 7412   99
```

```r
names(smart0)
```

```
 [1] "SEQNO"       "mmsa"        "mmsa_code"
 [4] "mmsa_name"   "mmsa_wt"     "completed"
 [7] "landline"    "hhadults"    "genhealth"
[10] "physhealth"  "menthealth"  "poorhealth"
[13] "agegroup"    "age_imp"     "race"
[16] "hispanic"    "race_eth"    "female"
[19] "marital"     "kids"        "educgroup"
[22] "home_own"    "veteran"     "employment"
[25] "incomegroup" "inc_imp"     "cell_own"
[28] "internet30"  "weight_kg"   "height_m"
[31] "bmi"         "bmigroup"    "pregnant"
[34] "deaf"        "blind"       "decide"
[37] "diffwalk"    "diffdress"   "diffalone"
```

For our In-Class Work . . .

```
smart1 <- smart0 %>%
    mutate(SEQNO = as.character(SEQNO)) %>%
    select(SEQNO, mmsa, mmsa_wt, landline,
           age_imp, healthplan, dm_status,
           fruit_day, drinks_wk, activity,
           smoker, physhealth, bmi, genhealth)

dim(smart1)
```

```
[1] 7412   14
```

Our 14 Variables in `smart1`

```r
str(smart1)
```

```
Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame':    74
 $ SEQNO     : chr  "2017000001" "2017000002" "2017000003" "20
 $ mmsa      : chr  "Cincinnati" "Cincinnati" "Cincinnati" "Ci
 $ mmsa_wt   : num  670 407 356 203 194 ...
 $ landline  : num  1 1 1 1 1 1 1 1 1 1 ...
 $ age_imp   : num  36 41 55 61 57 24 65 53 51 42 ...
 $ healthplan: num  1 1 1 1 1 0 1 1 1 1 ...
 $ dm_status : chr  "No-Diabetes" "No-Diabetes" "No-Diabetes" 
 $ fruit_day : num  1.43 1 3 0.5 0.72 2.5 3 0 0.14 NA ...
 $ drinks_wk : num  4.67 0 0 0 0.23 1.87 0 0 0.23 0 ...
 $ activity  : chr  "Active" NA "Highly_Active" "Insufficientl
 $ smoker    : chr  "Never" "Never" "Never" "Never" ...
 $ physhealth: num  0 0 2 0 2 0 0 30 2 30 ...
 $ bmi       : num  25.8 26.6 29.6 29.4 27.5 ...
 $ genhealth : chr  "2_VeryGood" "2_VeryGood" "2_VeryGood" "2_
```

Metropolitan Statistical Areas

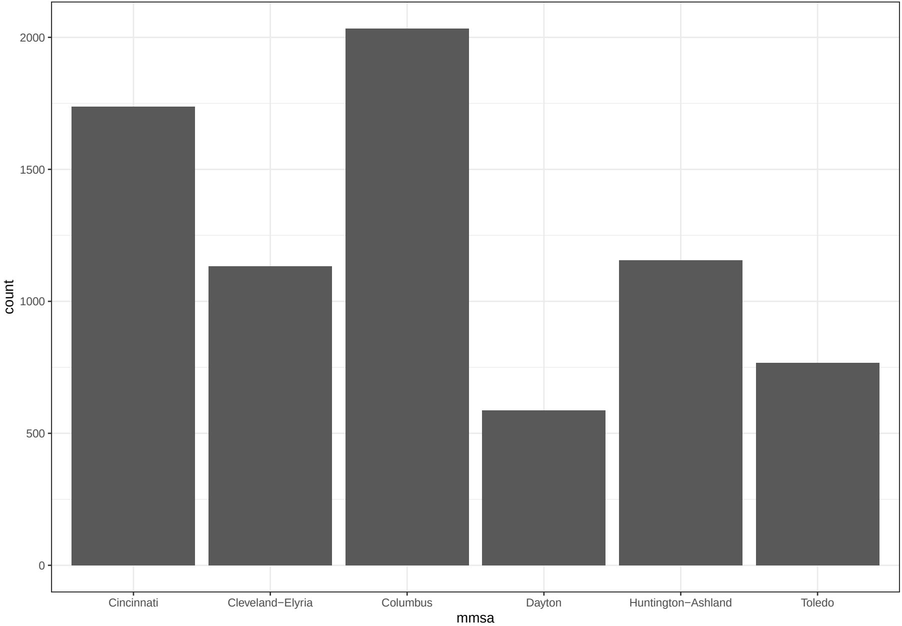
```r
smart1 %>% count(mmsa)
```

```
# A tibble: 6 x 2
  mmsa                  n
  <chr>             <int>
1 Cincinnati         1737
2 Cleveland-Elyria   1133
3 Columbus           2033
4 Dayton              587
5 Huntington-Ashland 1156
6 Toledo              766
```

Bar Chart, version 1 (code)

```r
ggplot(smart1, aes(x = mmsa)) +
  geom_bar()
```

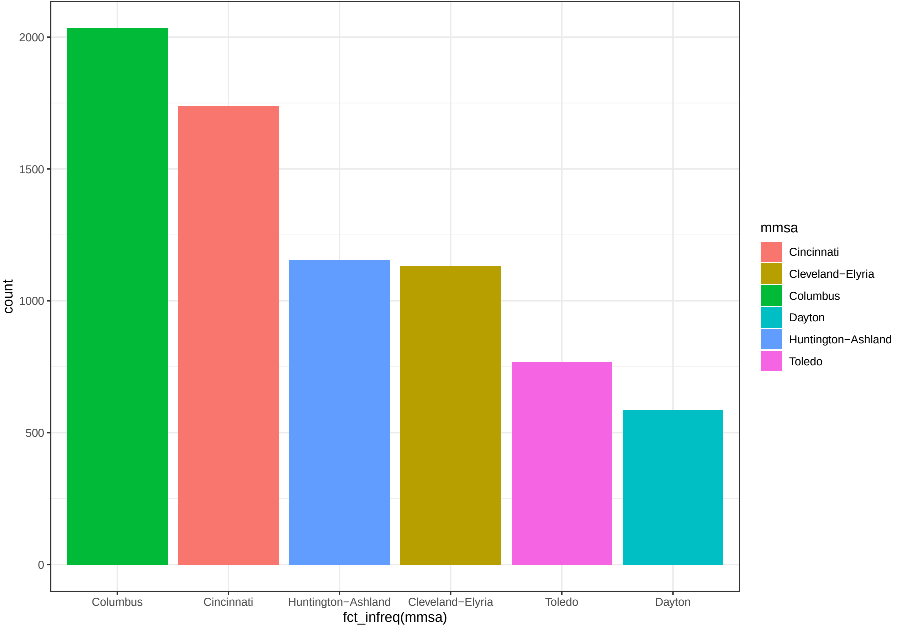
Bar Chart, version 1



Bar Chart, version 2 (code)

```r
ggplot(smart1, aes(x = fct_infreq(mmsa), fill = mmsa)) +
  geom_bar()
```

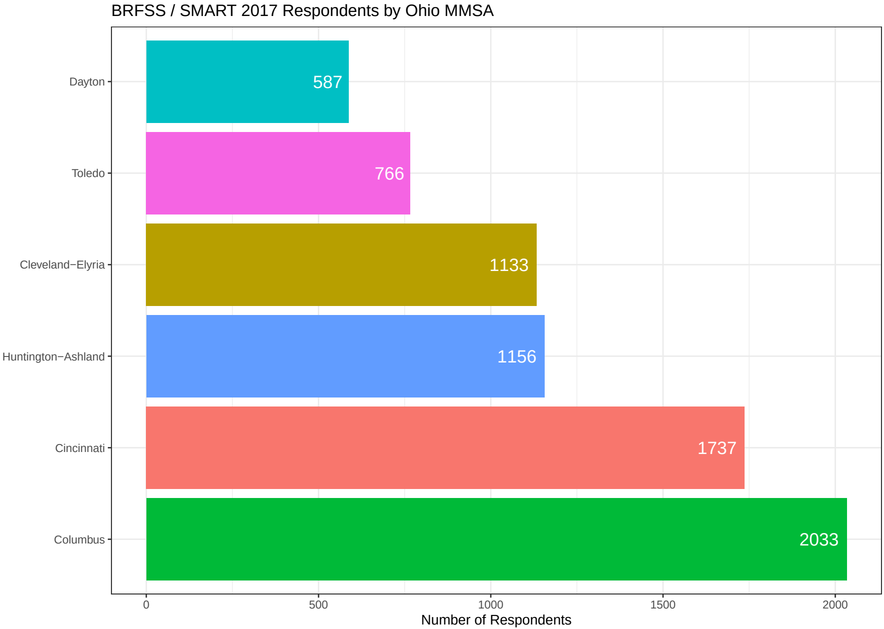
Bar Chart, version 2



Bar Chart, version 3 (code)

```
ggplot(smart1, aes(x = fct_infreq(mmsa), fill = mmsa)) +
  geom_bar() +
  geom_text(aes(label = ..count..), stat = "count",
            hjust = 1.2, size = 5, col = "white") +
  coord_flip() +
  guides(fill = FALSE) +
  labs(x = "",
       y = "Number of Respondents",
       title = "BRFSS / SMART 2017 Respondents by Ohio MMSA")
```

Bar Chart, version 3



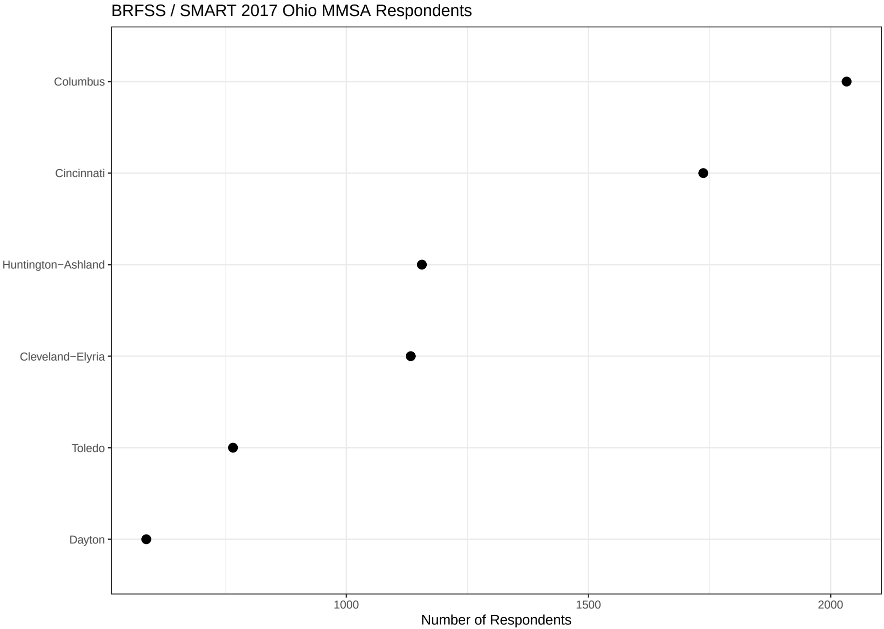
Cleveland Dot Plot (code)

```
smart1 %>% tabyl(mmsa)
```

```
              mmsa    n   percent
        Cincinnati 1737 0.2343497
  Cleveland-Elyria 1133 0.1528602
          Columbus 2033 0.2742849
            Dayton  587 0.0791959
Huntington-Ashland 1156 0.1559633
            Toledo  766 0.1033459
```

```
smart1 %>% tabyl(mmsa) %>%
  ggplot(., aes(x = n, y = reorder(mmsa, n))) +
  geom_point(size = 3) +
  labs(y = "",
       x = "Number of Respondents",
       title = "BRFSS / SMART 2017 Ohio MMSA Respondents")
```

Cleveland Dot Plot



Subject Identifiers

```
smart1 %>% select(SEQNO, mmsa_wt) %>% head()
```

```
# A tibble: 6 x 2
  SEQNO      mmsa_wt
  <chr>        <dbl>
1 2017000001    670.
2 2017000002    407.
3 2017000003    356.
4 2017000004    203.
5 2017000005    194.
6 2017000006    602.
```

Our Remaining Variables, by Type

Variable	Type	Description
`landline`	Binary (1/0)	survey conducted by landline? (vs. cell)
`healthplan`	Binary (1/0)	subject has health insurance?
`age_imp`	Quantitative	age (imputed from groups - see Notes)
`fruit_day`	Quantitative	mean servings of fruit / day
`drinks_wk`	Quantitative	mean alcoholic drinks / week
`bmi`	Quantitative	body-mass index (in kg/m^2)
`physhealth`	Count (0-30)	of last 30 days, # in poor physical health
`dm_status`	Categorical	diabetes status (4 levels)
`activity`	Categorical	physical activity level (4 levels)
`smoker`	Categorical	smoking status (4 levels)
`genhealth`	Categorical	self-reported overall health (5 levels)

The Art of Statistics

Chapter 1: Getting Things in Proportion: Categorical Data and Percentages

- Binary variables are yes/no questions, sets of which can be summarized as proportions.
- Positive or negative framing of proportions can change their emotional impact.
- Relative risks tend to convey an exaggerated importance, and absolute risks should be provided for clarity.
- Expected frequencies promote understanding and an appropriate sense of importance.
- Odds ratios arise from scientific studies but should not be used for general communication.
- Graphics need to be chosen with care and awareness of their impact.

Managing our Binary Variables

```
smart1 %>% count(landline)
```

```
# A tibble: 2 x 2
  landline     n
     <dbl> <int>
1        0  3763
2        1  3649
```

```
smart1 %>% tabyl(healthplan)
```

```
 healthplan    n     percent valid_percent
          0  398 0.053696708    0.05384199
          1 6994 0.943604965    0.94615801
         NA   20 0.002698327            NA
```

Can we impute the missing healthplan information?

Take a random draw from the existing distribution of healthplan?

```
set.seed(2020432)
smart1 <- smart1 %>%
    mutate(healthplan_i1 = healthplan) %>%
  data.frame() %>%
    impute_rhd(., healthplan_i1 ~ 1) %>%
  tbl_df()
```

- Why do we need the data.frame to tbl_df() shuffle here?

Simple imputation of `healthplan`: another option?

Use a model based on other (known) variables to impute `healthplan`?

```
set.seed(2020432)
smart1 <- smart1 %>%
    mutate(healthplan_i2 = factor(healthplan)) %>%
  data.frame() %>%
    impute_cart(., healthplan_i2 ~ landline + mmsa) %>%
  tbl_df()
```

- Why is it important to include `factor` here?

After simple imputation of `healthplan`

```
smart1 %>%
  count(healthplan, healthplan_i1, healthplan_i2)
```

```
# A tibble: 4 x 4
  healthplan healthplan_i1 healthplan_i2     n
       <dbl>         <dbl> <fct>         <int>
1          0             0 0               398
2          1             1 1              6994
3         NA             0 1                 1
4         NA             1 1                19
```

Was survey mode associated with healthplan?

Let's ignore the missing data for a moment...

```
sm1 <- smart1 %>%
  filter(complete.cases(landline, healthplan))

sm1 %>% tabyl(landline, healthplan)
```

```
 landline   0    1
        0 282 3473
        1 116 3521
```

Building a Better Table

```
sm1 %>% tabyl(landline, healthplan) %>%
  adorn_totals() %>%
  adorn_percentages() %>%
  adorn_pct_formatting() %>%
  adorn_ns(position = "front")
```

```
 landline            0             1
        0 282 (7.5%) 3473 (92.5%)
        1 116 (3.2%) 3521 (96.8%)
    Total 398 (5.4%) 6994 (94.6%)
```

Rearranging to form a useful 2 by 2 table

```
sm1 <- sm1 %>%
  mutate(insurance =
           fct_recode(factor(healthplan),
                      Insured = "1",
                      No_Ins = "0"),
         insurance = fct_relevel(insurance, "Insured"),
         style =
           fct_recode(factor(landline),
                      Land = "1",
                      Cell = "0"),
         style = fct_relevel(style, "Land"))

sm1 %$% table(style, insurance)
```

```
      insurance
style  Insured No_Ins
  Land    3521    116
  Cell    3473    282
```

Various 2x2 Table Analyses all at once...

```r
Epi::twoby2(sm1 %$% table(style, insurance),
            conf.level = 0.9)
```

```
2 by 2 table analysis:
------------------------------------------------------
Outcome   : Insured
Comparing : Land vs. Cell

     Insured No_Ins    P(Insured) 90% conf. interval
Land    3521    116        0.9681    0.9629   0.9726
Cell    3473    282        0.9249    0.9175   0.9317

                                    90% conf. interval
             Relative Risk: 1.0467    1.0372   1.0563
         Sample Odds Ratio: 2.4646    2.0470   2.9674
Conditional MLE Odds Ratio: 2.4644    2.0371   2.9907
    Probability difference: 0.0432    0.0347   0.0518
```

What's the best way to describe the results?

- Probability comparison?

96.8% of those reached by landline had insurance. 92.5% of those reached by cell phone had insurance.

- probability difference is 4.3 percentage points
- relative risk is 1.0467 (0.968/0.925)

Probability of having insurance was 4.67% higher among those contacted by landline.

What's the best way to describe the results?

- odds ratio = 2.4646

Those contacted by landline had almost 2.5 times the odds of having insurance as compared those contacted by cell phone.

- Difference in Expectation?

282 of the 3755 who answered by cell phone had no insurance. If the rate for those reached by landline applied to these people, too, then only 120 would have been expected to be without insurance.

Our Quantitative Variables

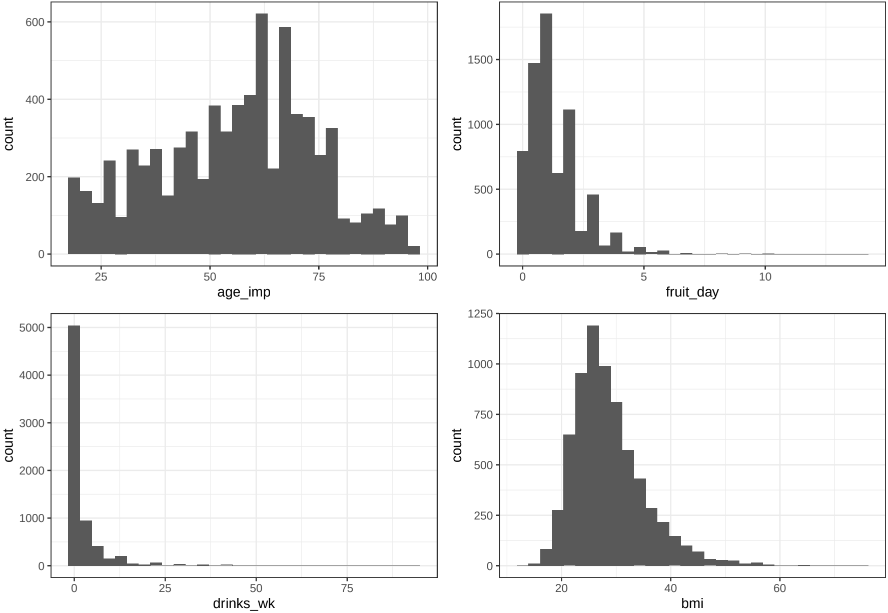
```r
smart1 %>%
  select(age_imp, fruit_day, drinks_wk, bmi) %>%
  mosaic::inspect()
```

```
quantitative variables:
       name   class  min    Q1 median    Q3   max      mean
1   age_imp numeric 18.0 42.00   58.0 69.00 96.00 55.932734
2 fruit_day numeric  0.0  0.57    1.0  2.00 14.00  1.340057
3 drinks_wk numeric  0.0  0.00    0.0  2.00 93.33  2.561651
4       bmi numeric 13.3 24.16   27.4 31.84 75.52 28.646485
         sd    n missing
1 18.413609 7344      68
2  1.122964 6855     557
3  6.564664 7020     392
4  6.616540 6919     493
```

Before we deal with the missingness... (code)

```
p_age <- ggplot(smart1, aes(x = age_imp)) +
  geom_histogram(bins = 30)

p_fru <- ggplot(smart1, aes(x = fruit_day)) +
  geom_histogram(bins = 30)

p_dri <- ggplot(smart1, aes(x = drinks_wk)) +
  geom_histogram(bins = 30)

p_bmi <- ggplot(smart1, aes(x = bmi)) +
  geom_histogram(bins = 30)

(p_age + p_fru) / (p_dri + p_bmi)
```

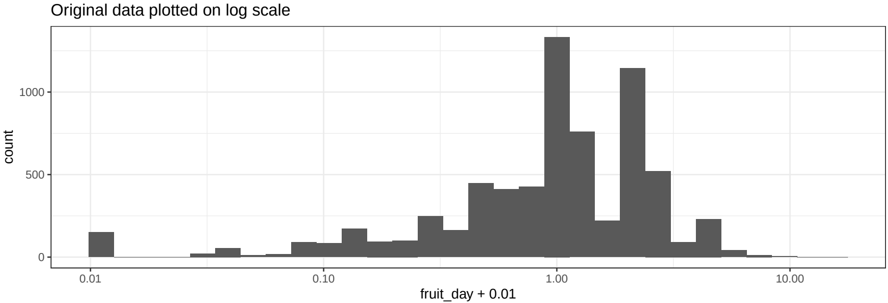
Histograms (suppressing NA warning message)

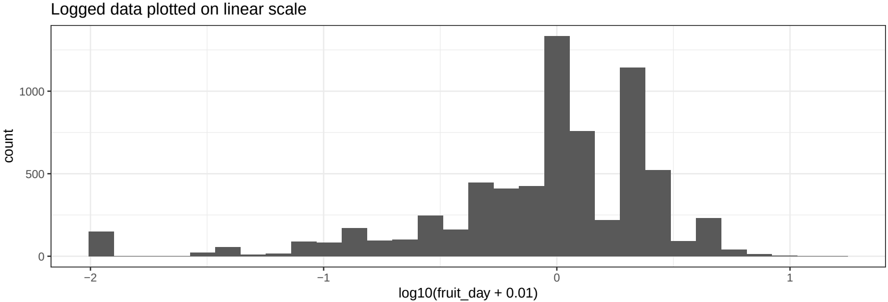


Should we put fruit_day on a log scale? (code)

```
p_1 <- ggplot(smart1, aes(x = fruit_day + 0.01)) +
  geom_histogram(bins = 30) +
  scale_x_log10() +
  labs(title = "Original data plotted on log scale")

p_2 <- ggplot(smart1, aes(x = log10(fruit_day + 0.01))) +
  geom_histogram(bins = 30) +
  labs(title = "Logged data plotted on linear scale")

p_1 / p_2
```

Should we put `fruit_day` on a log scale?





Simple Imputation of Quantities based on other variables?

```
set.seed(2020432)
smart1 <- smart1 %>%
    mutate(age_imp_i = age_imp,
           fruit_day_i = fruit_day,
           drinks_wk_i = drinks_wk,
           bmi_i = bmi) %>%
  data.frame() %>%
    impute_rlm(.,
                age_imp_i + fruit_day_i +
                  drinks_wk_i + bmi_i ~
                  mmsa + landline + healthplan_i1) %>%
  tbl_df()
```

Impact of Imputation here?

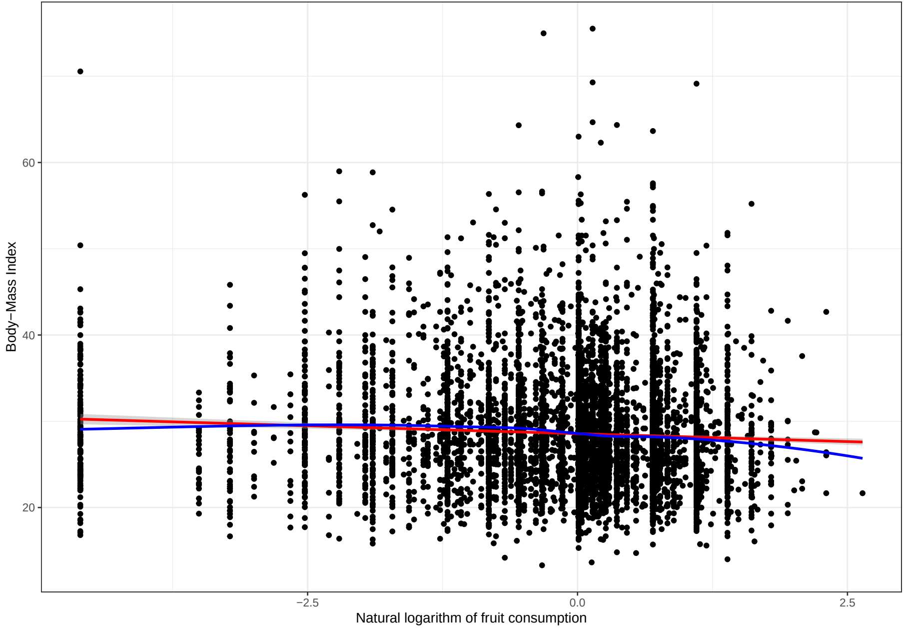
```
quantitative variables:
        name   class  min    Q1   median    Q3   max
1    age_imp numeric 18.0 42.00 58.00000 69.00 96.00
2  age_imp_i numeric 18.0 42.75 58.00000 69.00 96.00
3        bmi numeric 13.3 24.16 27.40000 31.84 75.52
4      bmi_i numeric 13.3 24.38 27.64954 31.41 75.52
      mean        sd    n missing
1 55.93273 18.413609 7344      68
2 55.93417 18.349847 7412       0
3 28.64649  6.616540 6919     493
4 28.60264  6.395698 7412       0
```

Is fruit consumption associated with BMI?

```
ggplot(smart1,
       aes(x = log(fruit_day_i + 0.01), y = bmi_i)) +
  geom_point() +
  geom_smooth(method = "lm", se = TRUE, col = "red") +
  geom_smooth(method = "loess", se = FALSE, col = "blue") +
  labs(x = "Natural logarithm of fruit consumption",
       y = "Body-Mass Index")
```

What do you think you'll see?

Is fruit consumption associated with BMI?



A Count (days of poor physical health in last 30)

```
a <- smart1 %>% tabyl(physhealth) %>% adorn_pct_formatting()
head(a, 3); tail(a, 3); rm(a)
```

```
 physhealth    n percent valid_percent
          0 4380   59.1%         60.2%
          1  311    4.2%          4.3%
          2  426    5.7%          5.9%
```

```
 physhealth   n percent valid_percent
         29  14    0.2%          0.2%
         30 677    9.1%          9.3%
         NA 138    1.9%             -
```

```
smart1 %$% mosaic::favstats(~ physhealth)
```

```
 min Q1 median Q3 max     mean       sd    n missing
   0  0      0  4  30 4.974842 9.408861 7274     138
```

Simple Imputation for physhealth based on bmi

```
set.seed(2020432)
smart1 <- smart1 %>%
    mutate(physhealth_i = physhealth) %>%
  data.frame() %>%
    impute_knn(., physhealth_i ~ bmi_i) %>%
  tbl_df()
```

- Why k-nearest neighbors here?

Results of imputation for `physhealth`

```r
a <- smart1 %>% filter(is.na(physhealth)) %>%
  tabyl(physhealth_i)

head(a, 3); tail(a, 3); rm(a)
```

```
 physhealth_i  n    percent
            0 93 0.67391304
            1  2 0.01449275
            2  8 0.05797101

 physhealth_i  n     percent
           25  1 0.007246377
           27  1 0.007246377
           30 18 0.130434783
```

Our Multi-Categorical Variables

```
smart1 %>%
  select(SEQNO, dm_status, activity, smoker, genhealth) %>%
  slice(201:204)
```

```
# A tibble: 4 x 5
  SEQNO      dm_status   activity      smoker       genhealth
  <chr>      <chr>       <chr>         <chr>        <chr>
1 2017000201 No-Diabetes Inactive      Never        3_Good
2 2017000202 No-Diabetes Highly_Acti~ Current_da~ 1_Excelle~
3 2017000203 Diabetes    Inactive      Former       2_VeryGood
4 2017000204 Diabetes    Inactive      Current_da~ 3_Good
```

What should we do here?

Using type.convert()

```
smart1 <- smart1 %>% type.convert()
smart1 %>%
  select(SEQNO, dm_status, activity, smoker, genhealth) %>%
  slice(431:432)
```

```
# A tibble: 2 x 5
       SEQNO dm_status   activity     smoker       genhealth
       <int> <fct>       <fct>        <fct>        <fct>
1 2017000431 No-Diabetes Highly_Acti~ Current_dai~ 4_Fair
2 2017000432 No-Diabetes Inactive     Never        5_Poor
```

- What does type.convert() do here?

`dm_status` is now a factor

```r
smart1 %>% tabyl(dm_status)
```

```
        dm_status    n     percent valid_percent
         Diabetes 1098 0.148138154   0.148418491
      No-Diabetes 6100 0.822989746   0.824547175
     Pre-Diabetes  133 0.017943875   0.017977832
 Pregnancy-Induced   67 0.009039396   0.009056502
             <NA>   14 0.001888829            NA
```

We could collapse to a binary (Yes/No) factor here...

```
smart1 <- smart1 %>%
  mutate(dm_f =
           fct_collapse(factor(dm_status),
                        Yes = "Diabetes",
                        No = c("No-Diabetes",
                               "Pre-Diabetes",
                               "Pregnancy-Induced")))
```

Simple Hot Deck Imputation for `dm_f`

```
set.seed(2020432)
smart1 <- smart1 %>%
    mutate(dm_f_i = dm_f) %>%
  data.frame() %>%
    impute_rhd(., dm_f_i ~ 1) %>%
  tbl_df()
```

Sanity Check

```
smart1 %>% count(dm_status, dm_f, dm_f_i)
```

```
Warning: Factor `dm_status` contains implicit NA, consider
using `forcats::fct_explicit_na`
```

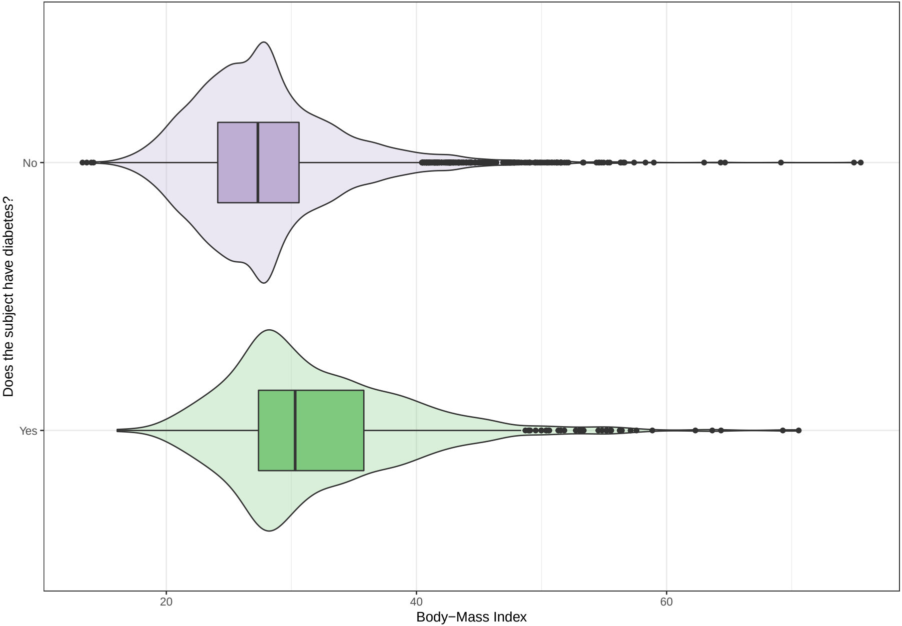
```
Warning: Factor `dm_f` contains implicit NA, consider using
`forcats::fct_explicit_na`
```

```
# A tibble: 6 x 4
  dm_status          dm_f  dm_f_i     n
  <fct>              <fct> <fct>  <int>
1 Diabetes           Yes   Yes     1098
2 No-Diabetes        No    No      6100
3 Pre-Diabetes       No    No       133
4 Pregnancy-Induced  No    No        67
5 <NA>               <NA>  Yes        3
6 <NA>               <NA>  No        11
```

Is diabetes status associated with BMI?

```
ggplot(smart1,
       aes(x = dm_f_i, y = bmi_i, fill = dm_f_i)) +
  geom_violin(alpha = 0.3) +
  geom_boxplot(width = 0.3) +
  scale_fill_brewer(type = "qual") +
  guides(fill = FALSE) +
  coord_flip() +
  labs(x = "Does the subject have diabetes?",
       y = "Body-Mass Index")
```

Is diabetes status associated with BMI?



smoker

```
smart1 %>% tabyl(smoker)
```

```
           smoker    n    percent valid_percent
    Current_daily  990 0.13356719     0.1380753
Current_not_daily  300 0.04047491     0.0418410
           Former 1999 0.26969779     0.2788006
            Never 3881 0.52361036     0.5412831
             <NA>  242 0.03264976            NA
```

Suppose we want to collapse the two "Current" categories together, and then impute?

Collapsing then imputing smoker into smoker_i

```
set.seed(2020432)
smart1 <- smart1 %>%
  mutate(smoker_f =
           fct_collapse(factor(smoker),
                        Current = c("Current_not_daily",
                                    "Current_daily")),
         smoker_i = smoker_f) %>%
  data.frame() %>%
    impute_rhd(., smoker_i ~ 1) %>%
  tbl_df()
```

Sanity Check

```
smart1 %>% tabyl(smoker, smoker_i)
```

```
           smoker Current Former Never
    Current_daily     990      0     0
Current_not_daily     300      0     0
           Former       0   1999     0
            Never       0      0  3881
             <NA>      50     61   131
```

activity

```
smart1 %>% tabyl(activity)
```

```
             activity    n    percent valid_percent
               Active 1132 0.15272531     0.1692331
        Highly_Active 2053 0.27698327     0.3069218
             Inactive 2211 0.29830005     0.3305427
 Insufficiently_Active 1293 0.17444684     0.1933024
                 <NA>  723 0.09754452            NA
```

What should we clean up here?

Imputing then Re-sorting the levels of `activity`

```
set.seed(2020432)
smart1 <- smart1 %>%
  mutate(activity_i = factor(activity)) %>%
  data.frame() %>%
    impute_rhd(., activity_i ~ 1) %>%
  tbl_df() %>%
  mutate(activity_i =
           fct_relevel(activity_i,
                       "Highly_Active",
                       "Active", "Insufficiently_Active",
                       "Inactive"))
```

Sanity Check

```
smart1 %>% count(activity_i, activity)
```

```
# A tibble: 8 x 3
  activity_i              activity                    n
  <fct>                   <fct>                   <int>
1 Highly_Active           Highly_Active            2053
2 Highly_Active           <NA>                      210
3 Active                  Active                   1132
4 Active                  <NA>                      124
5 Insufficiently_Active Insufficiently_Active      1293
6 Insufficiently_Active <NA>                        150
7 Inactive                Inactive                 2211
8 Inactive                <NA>                      239
```

genhealth

```r
smart1 %>% tabyl(genhealth)
```

```
   genhealth    n     percent valid_percent
 1_Excellent 1057 0.142606584     0.1428958
  2_VeryGood 2406 0.324608743     0.3252670
      3_Good 2367 0.319347005     0.3199946
      4_Fair 1139 0.153669725     0.1539813
      5_Poor  428 0.057744199     0.0578613
        <NA>   15 0.002023745            NA
```

Let's impute here with `activity_i`, `physhealth_i`, `mmsa` and `healthplan`

Simple Imputation of genhealth

```r
set.seed(2020432)
smart1 <- smart1 %>%
  mutate(genhealth_i = factor(genhealth)) %>%
  data.frame() %>%
    impute_cart(., genhealth_i ~ activity_i + physhealth_i +
                    factor(mmsa) + healthplan) %>%
  tbl_df()
```

Checking the Imputation's Impact

```
smart1 %>% tabyl(genhealth, genhealth_i)
```

```
   genhealth 1_Excellent 2_VeryGood 3_Good 4_Fair 5_Poor
 1_Excellent        1057          0      0      0      0
  2_VeryGood           0       2406      0      0      0
      3_Good           0          0   2367      0      0
      4_Fair           0          0      0   1139      0
      5_Poor           0          0      0      0    428
        <NA>           0         14      1      0      0
```

Fitting a Huge Regression Model

Without Imputation

```
model1 <- lm(bmi ~ mmsa + healthplan + age_imp + fruit_day +
               drinks_wk + physhealth + dm_f + activity +
               smoker_f + genhealth, data = smart1)
```

Using the Imputed Values

```
model1_i <- lm(bmi_i ~ mmsa + healthplan_i1 + age_imp_i +
               fruit_day_i + drinks_wk_i + physhealth_i +
               dm_f_i + activity_i + smoker_i +
               genhealth_i, data = smart1)
```

Compare the Two Models?

```
glance(model1) %>%
  select(r.squared, sigma, df, df.residual, AIC, BIC)
```

```
# A tibble: 1 x 6
  r.squared sigma    df df.residual    AIC    BIC
      <dbl> <dbl> <int>       <int>  <dbl>  <dbl>
1     0.122  6.24    21        5989 39093. 39241.
```

```
glance(model1_i) %>%
  select(r.squared, sigma, df, df.residual, AIC, BIC)
```

```
# A tibble: 1 x 6
  r.squared sigma    df df.residual    AIC    BIC
      <dbl> <dbl> <int>       <int>  <dbl>  <dbl>
1     0.109  6.05    21        7391 47734. 47886.
```

- Why are the `df` different?

From model1 (no imputation)

```
tidy(model1, conf.int = TRUE, conf.level = 0.9) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  slice(1:2)
```

```
# A tibble: 2 x 5
  term                   estimate std.error conf.low conf.high
  <chr>                     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)              29.7       0.605  28.7       30.7
2 mmsaCleveland-Elyria      0.420     0.267  -0.0191     0.858
```

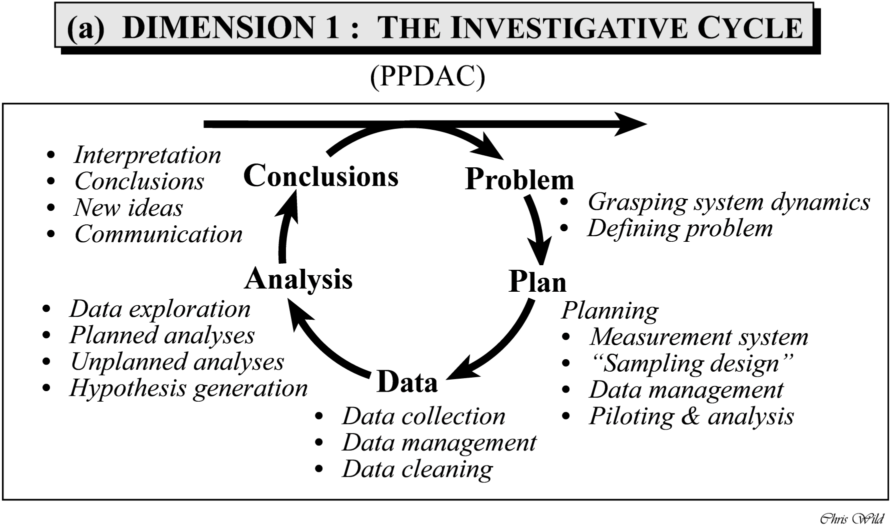
```
tidy(model1_i, conf.int = TRUE, conf.level = 0.9) %>%
  select(term, estimate, std.error, conf.low, conf.high) %>%
  slice(1:2)
```

```
# A tibble: 2 x 5
  term                   estimate std.error conf.low conf.high
  <chr>                     <dbl>     <dbl>    <dbl>     <dbl>
1 (Intercept)              29.4       0.519   28.5      30.2
2 mmsaCleveland-Elyria      0.275     0.231   -0.106     0.655
```

The Art of Statistics: How to Learn From Data

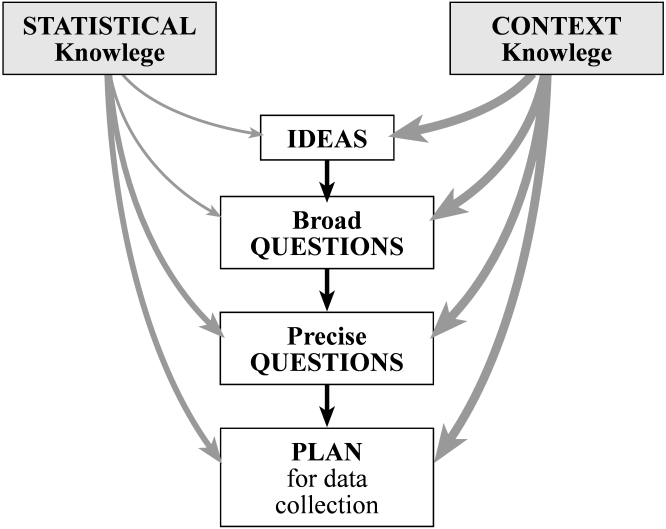
Introduction: Why We Need Statistics / Turning the World into Data

- Turning experiences into data is not straightforward, and data is inevitably limited in its capacity to describe the world.
- Statistical science has a long and successful history, but is now changing in the light of increased availability of data.
- Skill in statistical methods plays an important part of being a data scientist.
- Teaching statistics is changing from a focus on mathematical methods to one based on an entire problem-solving cycle.
- The PPDAC cycle provides a convenient framework…
    - Problem - Plan - Data - Analysis - Conclusion and communication.
- Data literacy is a key skill for the modern world.



- Chris Wild, https://www.stat.auckland.ac.nz/~wild/StatThink/

From inkling to plan



Chris Wild