SMI606: Week 2
Types of quantification

Dr. Calum Webb
Sheffield Methods Institute, the University of Sheffield.
c.j.webb@sheffield.ac.uk (mailto:c.j.webb@sheffield.ac.uk)

Sign In

Learning Objectives

What will I learn? (?panelset=what-will-i-learn%3F#panelset_what-will-i-learn%3F)

How does this week fit into my course? (?panelset=how-does-this-week-fit-into-my-course%3F#panelset_how-does-this-week-fit-into-my-course%3F)

By the end of this week you will:

- Be able to identify different variables using the continuous/ordinal/categorical typology.
- Be able to relate these variable types to variable classes in **R**.
- Know how a range of descriptive statistics can be used to summarise and simplify large amounts of data, and which ones to use for different types of variable.
- Be able to produce descriptive statistics and basic visualisations of different types of variables using **R**.

Learning Objectives

What will I learn? (?panelset=what-will-i-learn%3F#panelset_what-will-i-learn%3F)

How does this week fit into my course? (?panelset=how-does-this-week-fit-into-my-course%3F#panelset_how-does-this-week-fit-into-my-course%3F)

- Being able to interpret summary and descriptive statistics is a key skill in quantitative social science, with the first table of any research paper often being a table describing all variables.
- Social researchers regularly make use of descriptive statistics and visualisations to inform their analytical approach.
- Visualising distributions is an important part of checking for outliers, errors, and violated assumptions in statistical modelling.

Illustration by Allison Horst

Continuous and discrete variables in R

Continuous and discrete variables are largely handled through the **numeric** class that we saw in last week's practical.

```
continuous <- c(1.00735, 2.0873, 9.328, 12.4747)
class(continuous)
```

```
## [1] "numeric"
```

```
discrete <- c(1, 2, 3, 4)
class(discrete)
```

```
## [1] "numeric"
```

However, **R** does have a class of variable with specific functionality for discrete data: integer.

```
as.integer(continuous)
```

```
## [1]  1  2  9 12
```

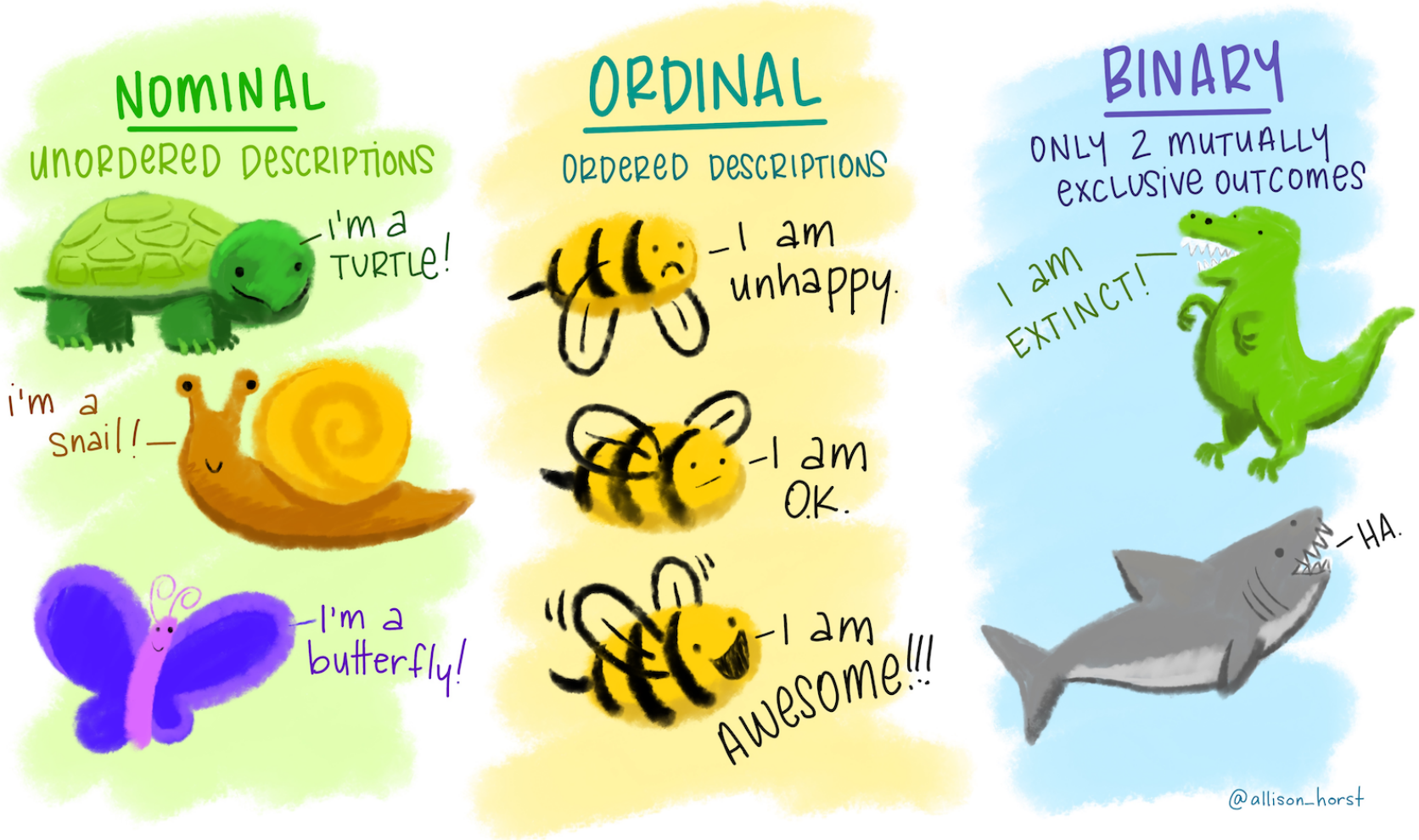


Illustration by Allison Horst

Nominal, Ordinal, and Binary Variables in R

There are multiple different ways to handle these kinds of variables in R. You can use simple character variables for nominal and binary variables:

```
nominal <- c("turtle", "snail", "butterfly")
class(nominal)
```

```
## [1] "character"
```

Nominal, Ordinal, and Binary Variables in R

You can use ordered factors for ordinal variables...

```
ordinal <- factor(c("unhappy", "ok", "awesome"), levels = c("unhappy", "ok", "awesome"),
                  ordered = TRUE)
print(ordinal)
```

```
## [1] unhappy ok      awesome
## Levels: unhappy < ok < awesome
```

```
class(ordinal)
```

```
## [1] "ordered" "factor"
```

... or simply make a numeric variable that corresponds to their character labels.

```
tibble(label = c("unhappy", "ok", "awesome"), numeric_val = c(1, 2, 3))
```

```
## # A tibble: 3 × 2
##   label   numeric_val
##   <chr>         <dbl>
## 1 unhappy           1
## 2 ok                2
## 3 awesome           3
```

Nominal, Ordinal, and Binary Variables in R

And lastly, we could store our binary variables as either numeric (identified by 0 = false or 1 = true); character; factor; or logical variables...

```
binary_num <- c(1, 0)
binary_chr <- c("extinct", "not extinct")
binary_fct <- factor(c("extinct", "not extinct"), levels = c("not extinct", "extinct"))
binary_lgc <- c(TRUE, FALSE)

tibble(binary_num, binary_chr, binary_fct, binary_lgc)
```

```
## # A tibble: 2 × 4
##   binary_num binary_chr  binary_fct  binary_lgc
##        <dbl> <chr>       <fct>       <lgl>
## 1          1 extinct     extinct     TRUE
## 2          0 not extinct not extinct FALSE
```

Why do we need to know 'types' of variable?

The kinds of statistical methods and data visualisations we need to use depends on the type of variable we are analysing/visualising.

Knowing this will also help us know how we need to tidy our data in `R` before we can analyse it.

Univariate Descriptive Statistics
Measures of Central Tendency

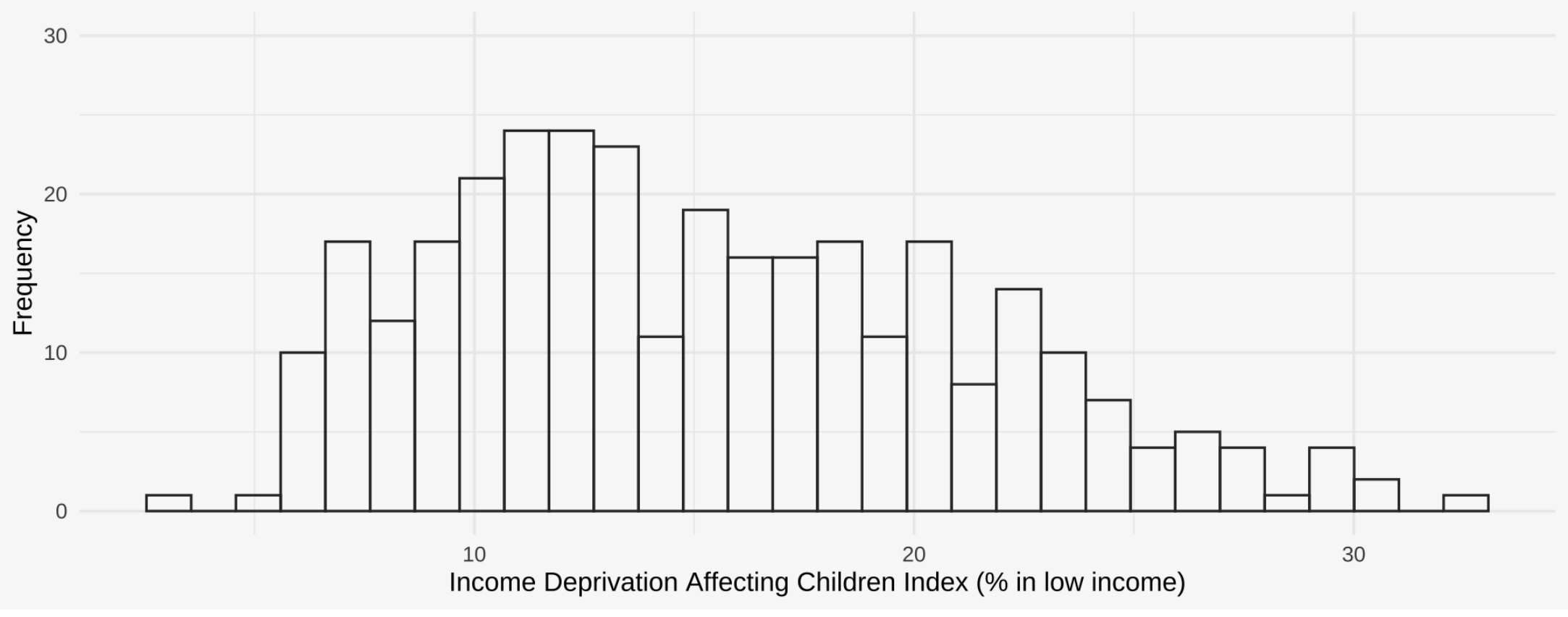
- Summarising data using the most frequent response.

How do we effectively communicate this?

Percentage of children in poverty

```
##   [1] 10.4 15.1 22.8 30.7 14.7  8.9 15.3 20.6
##   [9] 11.4 10.7 14.5 16.4 22.2 20.3 21.1 12.1
##  [17] 11.8 23.9 28.3 12.4 18.0  3.2 29.8 24.2
##  [25] 19.6 18.9 32.7 15.0 27.4 18.3 12.4 17.4
##  [33] 29.8 20.8 20.3 20.2 16.0 25.6  7.2 12.0
##  [41] 14.7 10.4 20.5 19.2 20.9 25.7 14.9 21.3
##  [49] 18.6 21.9 13.2  8.7 10.2  6.7  5.6 10.0
##  [57] 14.3 15.1 15.0 13.6 21.9 15.4  8.9 10.9
##  [65] 23.8 12.6 22.5 19.9 19.7  9.9 16.4 16.0
##  [73] 27.6  8.8 20.4 21.9 16.1 22.5 12.2 12.9
##  [81] 18.2  8.9  8.6 13.1  9.5 15.5 13.0 25.5
##  [89] 16.9 19.6 12.2 19.3 17.2 14.7 14.9 15.4
##  [97] 12.2 10.8 13.0 10.5 20.2 10.7  6.1 12.0
## [105]  7.1 14.8 16.4 18.1  8.3 21.8  6.9 15.6
## [113] 18.5 11.9 13.7 10.1  8.7 22.7 19.4 20.6
## [121] 16.6  8.3 10.4  7.7  7.6 22.0 11.1 13.2
## [129] 15.0 19.3  8.5  8.7  6.9 22.5 12.4  7.3
## [137] 17.1 13.6  7.6 20.1 20.1 14.0 10.9 20.4
## [145] 13.8 17.4 15.8 17.2 24.8 21.2  8.4 24.9
## [153]  8.6 18.6  6.7 20.8 18.6  6.7 12.3  5.5
## [161] 26.5 18.5 16.0 10.8 12.2 15.7 11.0  6.6
## [169] 17.0 10.7 21.9 19.0 27.5 12.9 15.2 15.9
## [177] 10.0 17.7 30.3 22.9 17.5 20.3 12.8 22.9
## [185] 11.2 23.1 29.9 13.4 11.8 12.8 29.7 21.2
## [193] 10.6 12.3 13.4 11.6  9.8  6.5  6.4 10.8
```

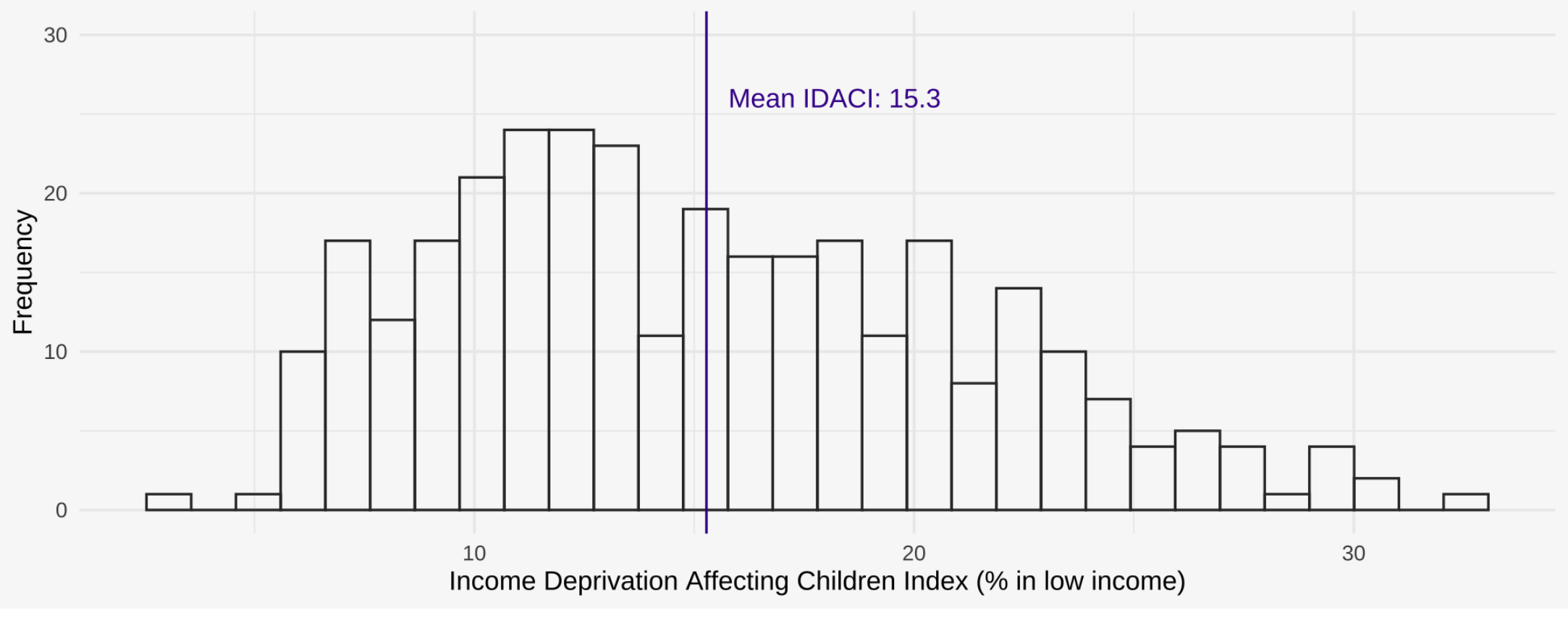
Univariate Descriptive Statistics
Measures of Central Tendency



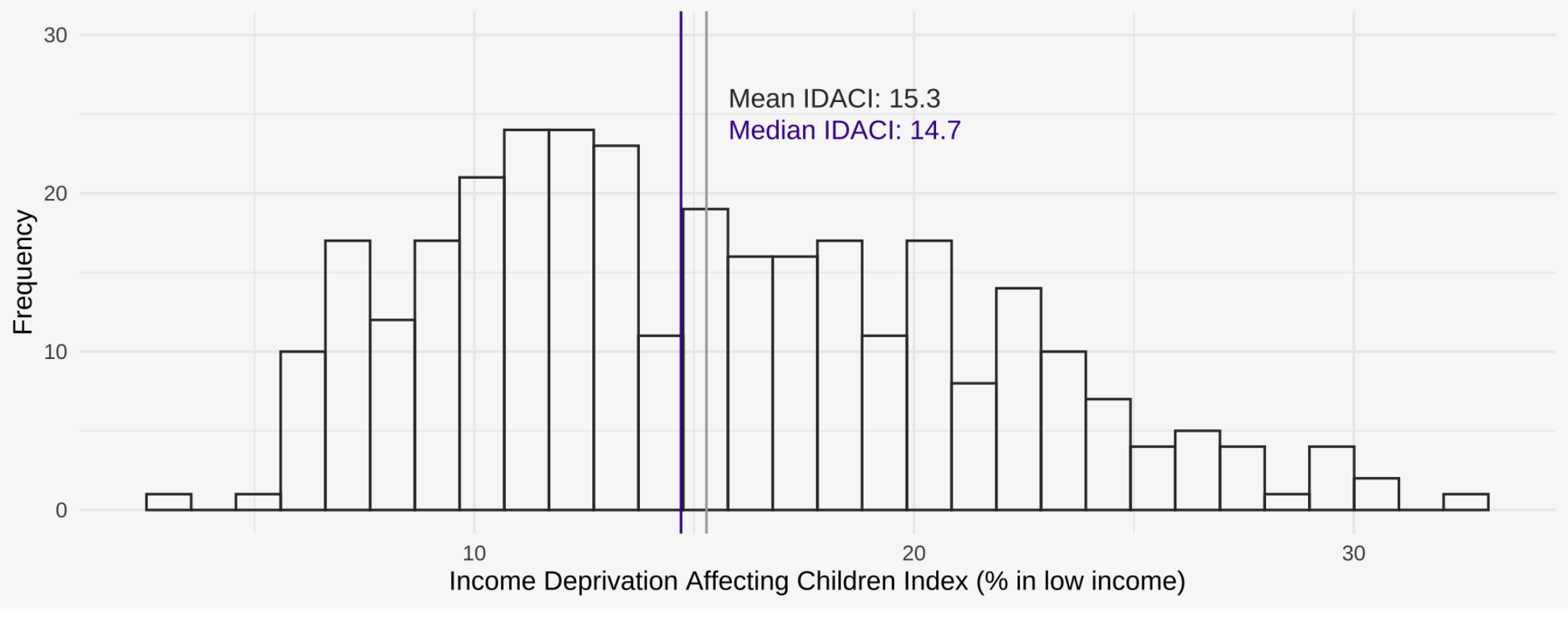
Side note: How did I create this plot?

```
idaci %>% # My data
  ggplot() + # start up ggplot
  geom_histogram(aes(x = idaci), col = "grey20", fill = "transparent") + # create a histogram
  theme_minimal() + # make a nice background
  xlab("Income Deprivation Affecting Children Index (% in low income)") + # Add an x axis label
  ylab("Frequency") # add a y axis label
```

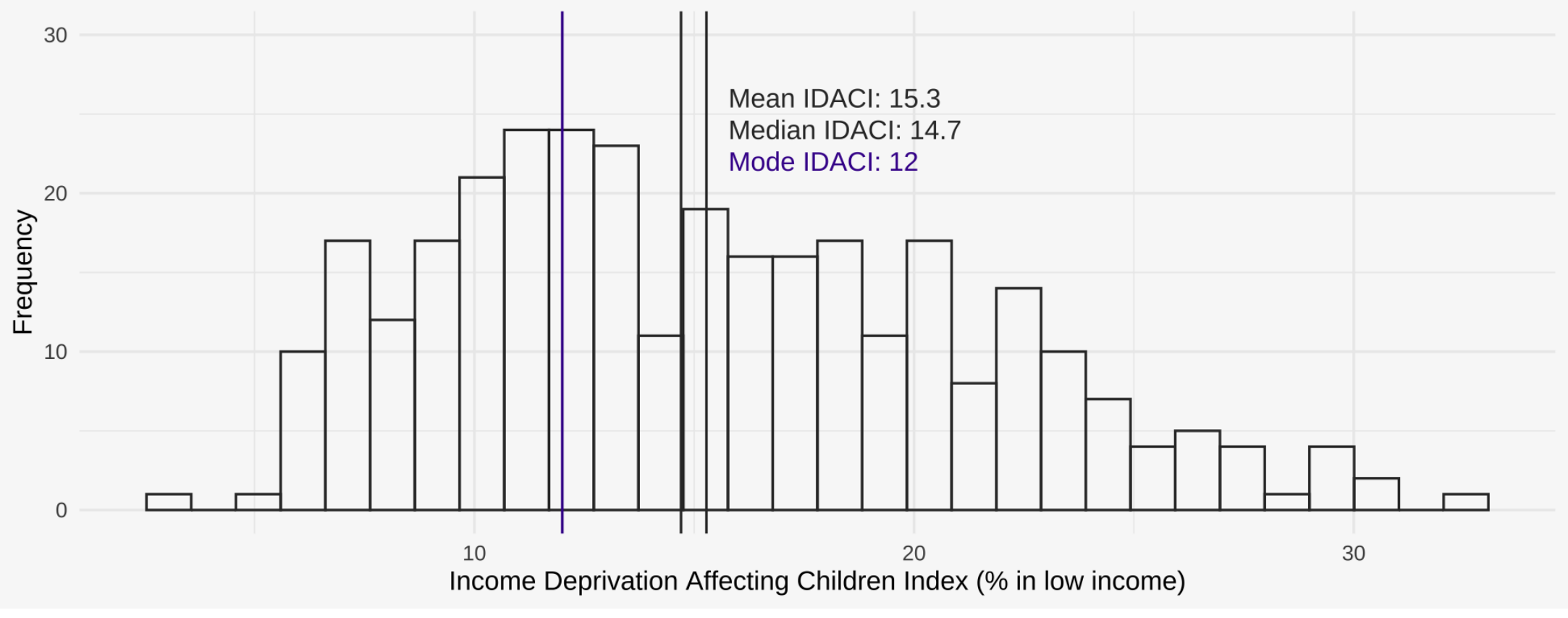
Univariate Descriptive Statistics
Measures of Central Tendency



Univariate Descriptive Statistics
Measures of Central Tendency



Univariate Descriptive Statistics
Measures of Central Tendency



Univariate Descriptive Statistics
Measures of Central Tendency

```
library(modeest) # Package for calculating the mode

hse_cleaned %>%
  summarise(
    height_mean =   mean(htval,    na.rm = TRUE), # Height
    topqual3_mode = mfv(topqual3,  na.rm = TRUE), # Qualifications
    eqvinc_mean =   mean(eqvinc,   na.rm = TRUE), # Income
    eqvinc_median = median(eqvinc, na.rm = TRUE)  # Income
            )
```

```
## # A tibble: 1 × 4
##   height_mean topqual3_mode    eqvinc_mean eqvinc_median
##         <dbl> <fct>                  <dbl>         <dbl>
## 1        168. No qualification      33274.        23443.
```

Univariate Descriptive Statistics
Measures of Central Tendency

```
library(modeest)

hse_cleaned %>%
  summarise(
    height_mean =   mean(htval,    na.rm = TRUE), # Height
    topqual3_mode = mfv(topqual3,  na.rm = TRUE), # Qualifications
    eqvinc_mean =   mean(eqvinc,   na.rm = TRUE), # Income
    eqvinc_median = median(eqvinc, na.rm = TRUE)  # Income
            )
```

```
## # A tibble: 1 × 4
##   height_mean topqual3_mode    eqvinc_mean eqvinc_median
##         <dbl> <fct>                  <dbl>         <dbl>
## 1        168. No qualification      33274.        23443.
```

Univariate Descriptive Statistics
Measures of Central Tendency

```
library(modeest)

hse_cleaned %>%
  summarise(
    height_mean =   mean(htval,    na.rm = TRUE), # Height
    topqual3_mode = mfv(topqual3,  na.rm = TRUE), # Qualifications
    eqvinc_mean =   mean(eqvinc,   na.rm = TRUE), # Income
    eqvinc_median = median(eqvinc, na.rm = TRUE)  # Income
            )
```

```
## # A tibble: 1 × 4
##   height_mean topqual3_mode    eqvinc_mean eqvinc_median
##         <dbl> <fct>                  <dbl>         <dbl>
## 1        168. No qualification      33274.        23443.
```

Univariate Descriptive Statistics
Measures of Central Tendency

```
library(modeest)

hse_cleaned %>%
  summarise(
    height_mean =   mean(htval,    na.rm = TRUE), # Height
    topqual3_mode = mfv(topqual3,  na.rm = TRUE), # Qualifications
    eqvinc_mean =   mean(eqvinc,   na.rm = TRUE), # Income
    eqvinc_median = median(eqvinc, na.rm = TRUE)  # Income
            )
```

```
## # A tibble: 1 × 4
##   height_mean topqual3_mode    eqvinc_mean eqvinc_median
##         <dbl> <fct>                  <dbl>         <dbl>
## 1        168. No qualification      33274.        23443.
```

Univariate Descriptive Statistics
Measures of Central Tendency

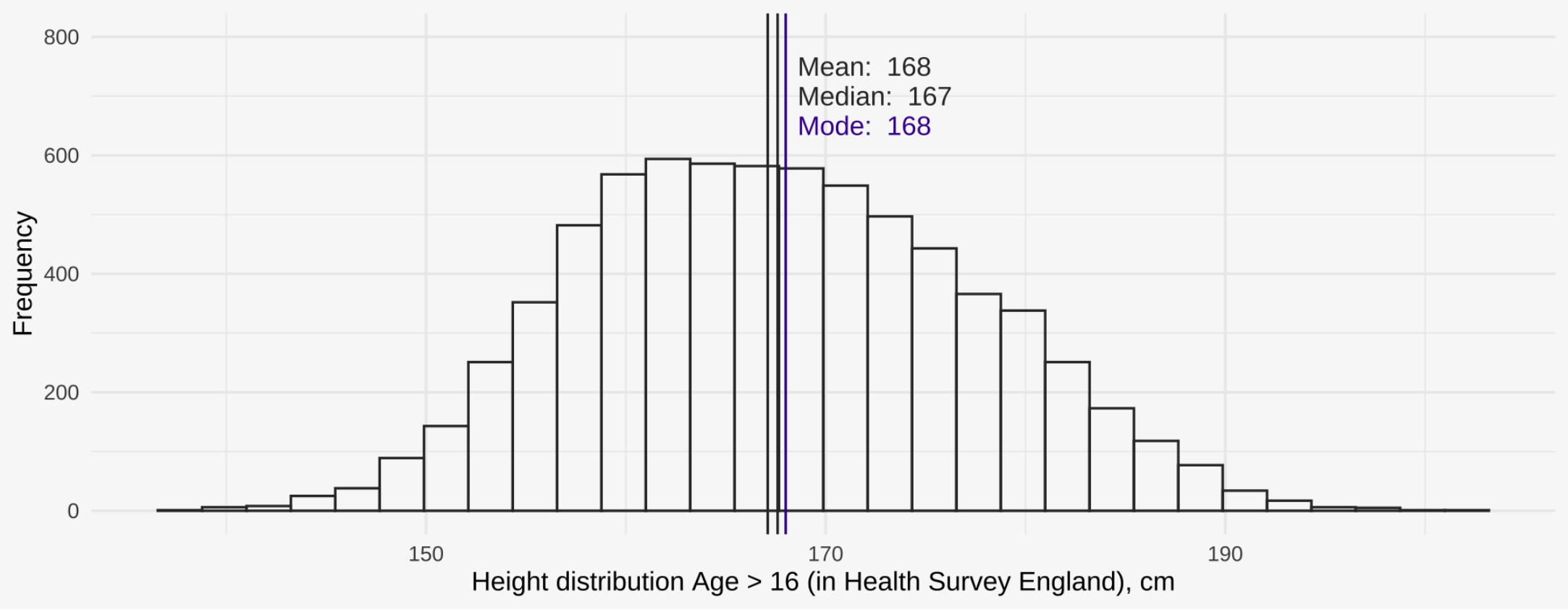
```
library(modeest)

hse_cleaned %>%
  summarise(
    height_mean =   mean(htval,    na.rm = TRUE), # Height
    topqual3_mode = mfv(topqual3,  na.rm = TRUE), # Qualifications
    eqvinc_mean =   mean(eqvinc,   na.rm = TRUE), # Income
    eqvinc_median = median(eqvinc, na.rm = TRUE)  # Income
            )
```

```
## # A tibble: 1 × 4
##   height_mean topqual3_mode    eqvinc_mean eqvinc_median
##         <dbl> <fct>                  <dbl>         <dbl>
## 1        168. No qualification      33274.        23443.
```

Univariate Descriptive Statistics
Measures of Central Tendency

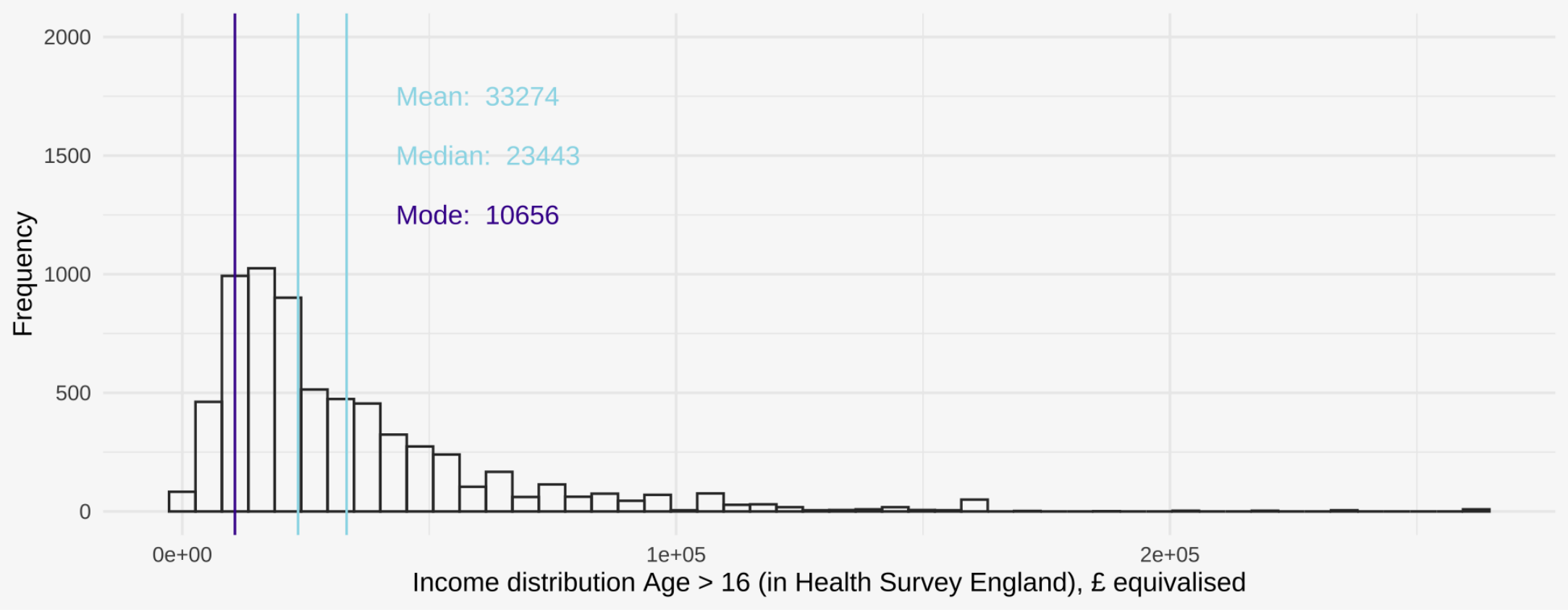
Univariate Descriptive Statistics
Measures of Central Tendency

Univariate Descriptive Statistics
Measures of Central Tendency



When a continuous/integer variable is normally distributed, the mean, median, and mode should all agree.

Univariate Descriptive Statistics
Measures of Central Tendency




are your
summary statistics
hiding something
interesting?
@allison_horst

When a continuous/integer variable is not normally distributed (for example, if it is skewed), the mean, median, and mode will often not align and some will not end up on the 'most likely value'.

This is why it's always important to visualise your data.

Univariate Descriptive Statistics
Measures of Central Tendency

- Summarising data using the most frequent response: categorical and ordinal variables.

How do we effectively communicate this?

Self-rated health

```
##  [1] 3. Good/Very Good 3. Good/Very Good 3. Good/Very Good
1. Bad/Very Bad
##  [5] 3. Good/Very Good 3. Good/Very Good 3. Good/Very Good
3. Good/Very Good
##  [9] 3. Good/Very Good 3. Good/Very Good 2. Fair
3. Good/Very Good
## [13] 1. Bad/Very Bad   3. Good/Very Good 3. Good/Very Good
3. Good/Very Good
## [17] 3. Good/Very Good 2. Fair           1. Bad/Very Bad
3. Good/Very Good
## [21] 3. Good/Very Good 3. Good/Very Good 1. Bad/Very Bad
1. Bad/Very Bad
## [25] 2. Fair           3. Good/Very Good 3. Good/Very Good
2. Fair
## [29] 2. Fair           2. Fair           2. Fair
2. Fair
## [33] 3. Good/Very Good 3. Good/Very Good 2. Fair
3. Good/Very Good
## [37] 3. Good/Very Good 3. Good/Very Good 2. Fair
1. Bad/Very Bad
## [41] 2. Fair           2. Fair           3. Good/Very Good
3. Good/Very Good
## [45] 3. Good/Very Good 3. Good/Very Good 2. Fair
3. Good/Very Good
## [49] 3. Good/Very Good 1. Bad/Very Bad
## Levels: 1. Bad/Very Bad 2. Fair 3. Good/Very Good
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about nominal/categorical variables?

1. For more than two categories, we can still use the mode.
2. For two categories stored as a logical variable (a binary variable), we can use the mean to calculate the proportion of each category.
3. We can use this same logic to create the proportion of all possible options in the categories (your typical percentages).

```
mfv(hse_cleaned$econact, na.rm = TRUE)
```

```
## [1] In employment
## 10 Levels: Refused ... Other economically inactive
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about nominal/categorical variables?

1. For more than two categories, we can still use the mode.
2. For two categories stored as a logical variable (a binary variable), we can use the mean to calculate the proportion of each category.
3. We can use this same logic to create the proportion of all possible options in the categories (your typical percentages).

```
hse_cleaned <- hse_cleaned %>%
  mutate(
    employed_lgc = ifelse(
      test = employed == "1. Employed",
      yes = TRUE,
      no = FALSE)
  )

mean(hse_cleaned$employed_lgc, na.rm = TRUE)
```

```
## [1] 0.545562
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about nominal/categorical variables?

1. For more than two categories, we can still use the mode.
2. For two categories stored as a logical variable (a binary variable), we can use the mean to calculate the proportion of each category.
3. We can use this same logic to create the proportion of all possible options in the categories (your typical percentages).

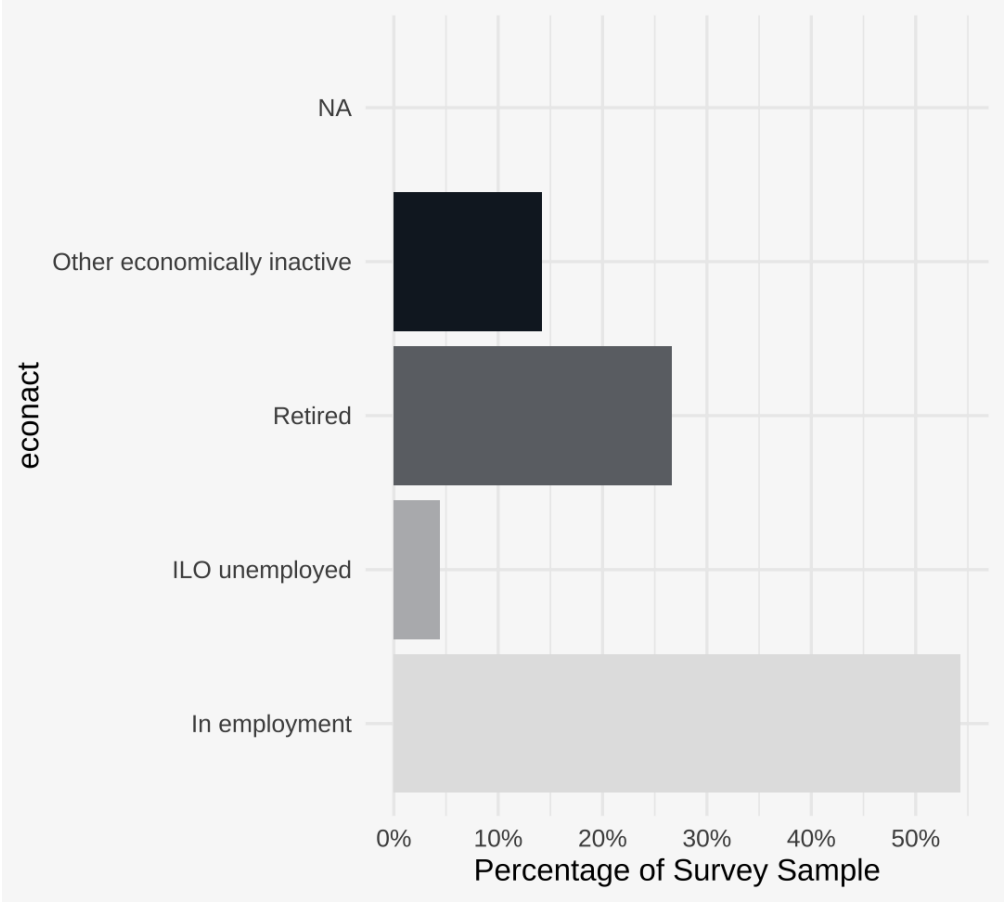
```
options(width = 40)

hse_cleaned %>%
  sjmisc::to_dummy("econact",
                   suffix = "label") %>%
  janitor::clean_names() %>%
  summarise_all(~mean(., na.rm = TRUE))
```

```
##   econact_in_employment
## 1              0.545562
##   econact_ilo_unemployed
## 1             0.04443919
##   econact_retired
## 1       0.2676989
##   econact_other_economically_inactive
## 1                                0.1423
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about nominal/categorical variables?

1. For more than two categories, we can still use the mode.
2. For two categories stored as a logical variable (a binary variable), we can use the mean to calculate the proportion of each category.
3. We can use this same logic to create the proportion of all possible options in the categories (your typical percentages).



Aside: How did I create this bar plot in R?

```
hse_cleaned %>% # my data
  ggplot(aes(econact)) + # the variable I want to create a bar plot of
  geom_bar(aes(y = ..count../sum(..count..), fill = econact)) + # bar plot as a percentage
  scale_y_continuous(labels=scales::percent_format()) + # formatting my axis as percentages
  theme_minimal() + # change the theme to one I think looks nice
  coord_flip() + # make it a horizontal bar plot
  ylab("Percentage of Survey Sample") + # an informative y label
  theme(legend.position = "none") # hide the legend it creates by default
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about nominal/categorical variables?

1. For more than two categories, we can still use the mode.
2. For two categories stored as a logical variable (a binary variable), we can use the mean to calculate the proportion of each category.
3. We can use this same logic to create the proportion of all possible options in the categories (your typical percentages).
4. Alternatively, we can use a simple table of **frequency counts** — best practice is usually to show **both** frequency counts **and** percentages, in case our sample is small.

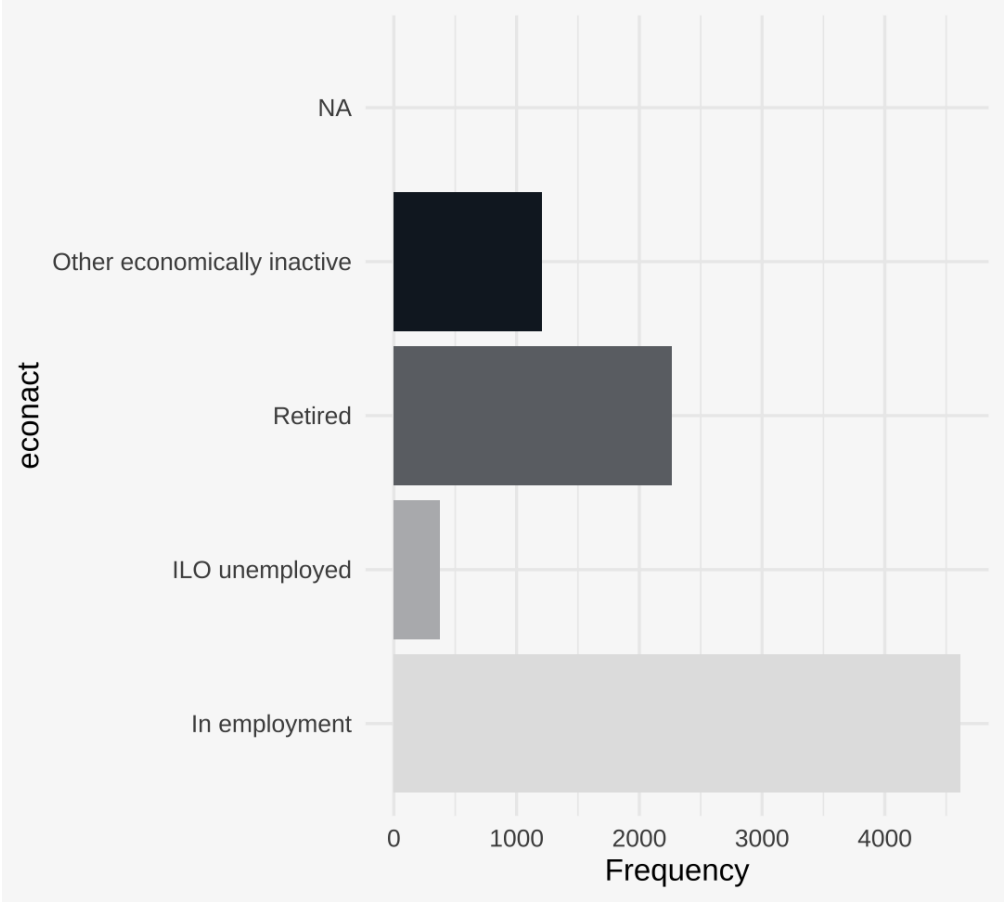
```
library(janitor)

tabyl(hse_cleaned$econact_s) %>%
  adorn_rounding(digits = 3)
```

```
##  hse_cleaned$econact_s    n percent
##               Employed 4616   0.543
##               Inactive 1204   0.142
##                Retired 2265   0.266
##                   <NA>  418   0.049
##  valid_percent
##          0.571
##          0.149
##          0.280
##             NA
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about nominal/categorical variables?

1. For more than two categories, we can still use the mode.
2. For two categories stored as a logical variable (a binary variable), we can use the mean to calculate the proportion of each category.
3. We can use this same logic to create the proportion of all possible options in the categories (your typical percentages).
4. Alternatively, we can use a simple table of **frequency counts** — best practice is usually to show **both** frequency counts **and** percentages, in case our sample is small.



Aside: How did I create this bar plot in R?

```
hse_cleaned %>%
  ggplot(aes(econact)) +
  geom_bar(aes(y = ..count.., fill = econact)) + # no longer a percentage
  theme_minimal() +
  coord_flip() +
  ylab("Frequency") +
  theme(legend.position = "none")
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about ordinal variables?

1. You can still use the mean, but it is unlikely to give a 'valid' value.
2. You can use the median for the numeric equivalent for the value.
3. You can use the mode for either the character equivalent or the numeric equivalent.

```
hse_cleaned %>%
  mutate(
    health_nm = as.numeric(health)
  ) %>%
  summarise(
    health_mean = mean(health_nm,
                       na.rm = TRUE),
    health_med = median(health_nm,
                        na.rm = TRUE),
    health_mfv = mfv(health,
                     na.rm = TRUE)
  )
```

```
## # A tibble: 1 × 3
##   health_mean health_med health_mfv
##         <dbl>      <dbl> <fct>
## 1        2.67          3 3. Good/Very G…
```

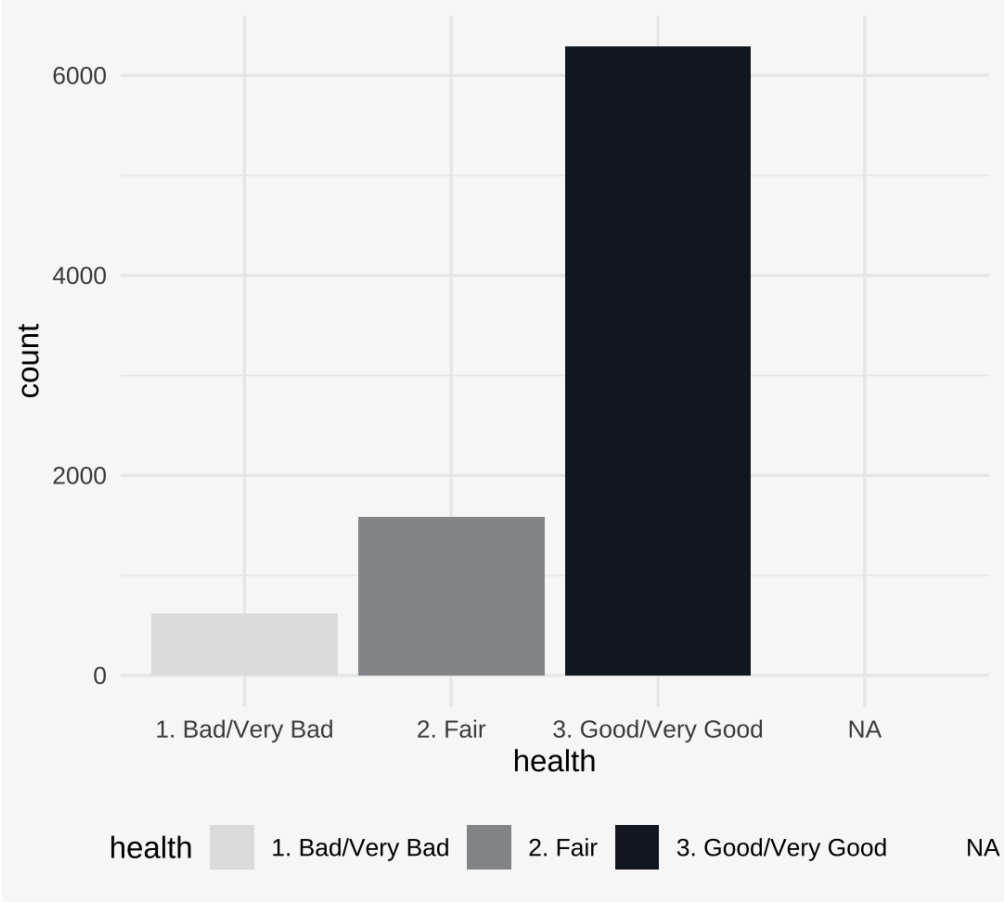
```
unique(hse_cleaned$health)
```

```
## [1] 3. Good/Very Good 1. Bad/Very Bad
## [3] 2. Fair           <NA>
## 3 Levels: 1. Bad/Very Bad ... 3. Good/Very Good
```

Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about ordinal variables?

1. You can still use the mean, but it is unlikely to give a 'valid' value.
2. You can use the median for the numeric equivalent for the value.
3. You can use the mode for either the character equivalent or the numeric equivalent.



Univariate Descriptive Statistics
Measures of Central Tendency/Summary Statistics

What about ordinal variables?

1. You can still use the mean, but it is unlikely to give a 'valid' value.
2. You can use the median for the numeric equivalent for the value.
3. You can use the mode for either the character equivalent or the numeric equivalent.
4. As before, a table is also an appropriate way to summarise the variable.

```
tabyl(hse_cleaned$health) %>%
  adorn_rounding(digits = 3)
```

```
##  hse_cleaned$health    n percent
##     1. Bad/Very Bad  620   0.073
##             2. Fair 1588   0.187
##   3. Good/Very Good 6288   0.740
##                <NA>    7   0.001
##  valid_percent
##          0.073
##          0.187
##          0.740
##             NA
```

Univariate Descriptive Statistics
Measures of Dispersion

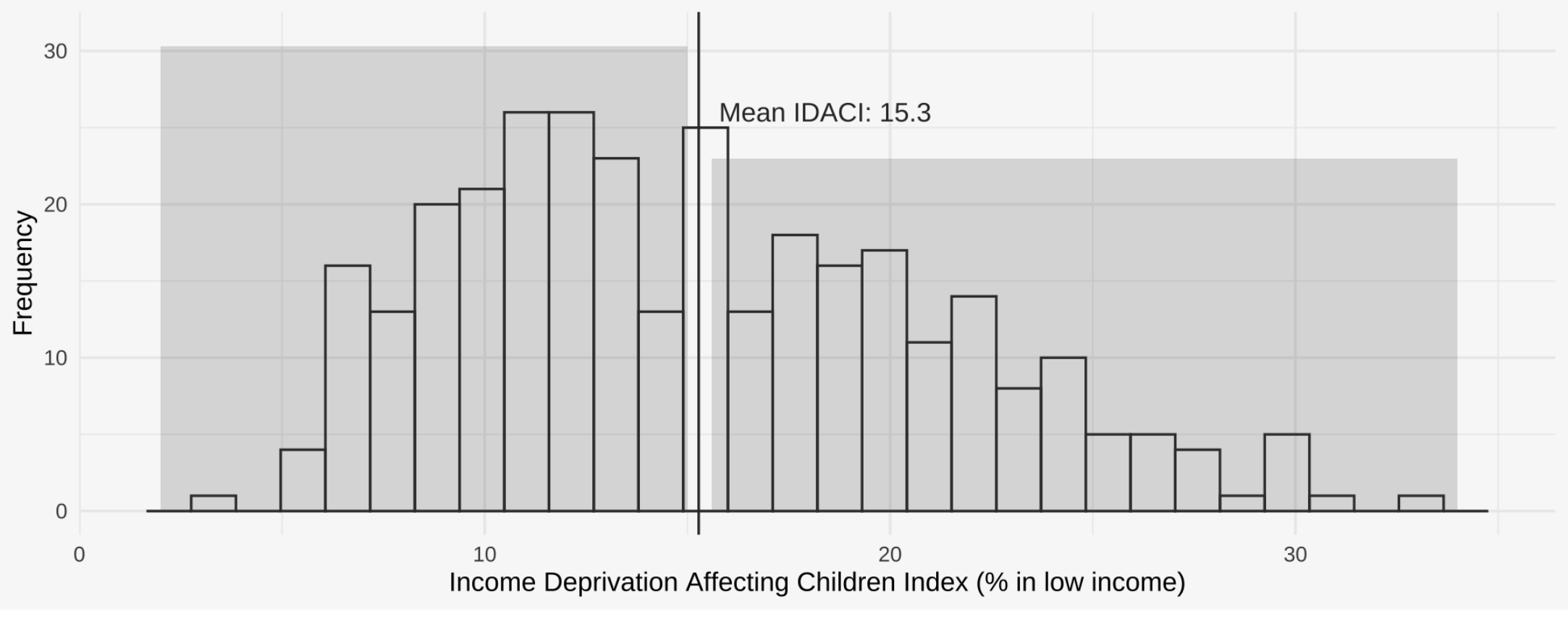
- We've summarised the most likely outcomes: but how can we effectively describe the range of responses in continuous data?

How do we effectively communicate the variation?

Percentage of children in poverty

```
##   [1] 10.4 15.1 22.8 30.7 14.7  8.9 15.3 20.6
##   [9] 11.4 10.7 14.5 16.4 22.2 20.3 21.1 12.1
##  [17] 11.8 23.9 28.3 12.4 18.0  3.2 29.8 24.2
##  [25] 19.6 18.9 32.7 15.0 27.4 18.3 12.4 17.4
##  [33] 29.8 20.8 20.3 20.2 16.0 25.6  7.2 12.0
##  [41] 14.7 10.4 20.5 19.2 20.9 25.7 14.9 21.3
##  [49] 18.6 21.9 13.2  8.7 10.2  6.7  5.6 10.0
##  [57] 14.3 15.1 15.0 13.6 21.9 15.4  8.9 10.9
##  [65] 23.8 12.6 22.5 19.9 19.7  9.9 16.4 16.0
##  [73] 27.6  8.8 20.4 21.9 16.1 22.5 12.2 12.9
##  [81] 18.2  8.9  8.6 13.1  9.5 15.5 13.0 25.5
##  [89] 16.9 19.6 12.2 19.3 17.2 14.7 14.9 15.4
##  [97] 12.2 10.8 13.0 10.5 20.2 10.7  6.1 12.0
## [105]  7.1 14.8 16.4 18.1  8.3 21.8  6.9 15.6
## [113] 18.5 11.9 13.7 10.1  8.7 22.7 19.4 20.6
## [121] 16.6  8.3 10.4  7.7  7.6 22.0 11.1 13.2
## [129] 15.0 19.3  8.5  8.7  6.9 22.5 12.4  7.3
## [137] 17.1 13.6  7.6 20.1 20.1 14.0 10.9 20.4
## [145] 13.8 17.4 15.8 17.2 24.8 21.2  8.4 24.9
## [153]  8.6 18.6  6.7 20.8 18.6  6.7 12.3  5.5
## [161] 26.5 18.5 16.0 10.8 12.2 15.7 11.0  6.6
## [169] 17.0 10.7 21.9 19.0 27.5 12.9 15.2 15.9
## [177] 10.0 17.7 30.3 22.9 17.5 20.3 12.8 22.9
## [185] 11.2 23.1 29.9 13.4 11.8 12.8 29.7 21.2
## [193] 10.6 12.3 13.4 11.6  9.8  6.5  6.4 10.8
```

Univariate Descriptive Statistics
Measures of Dispersion

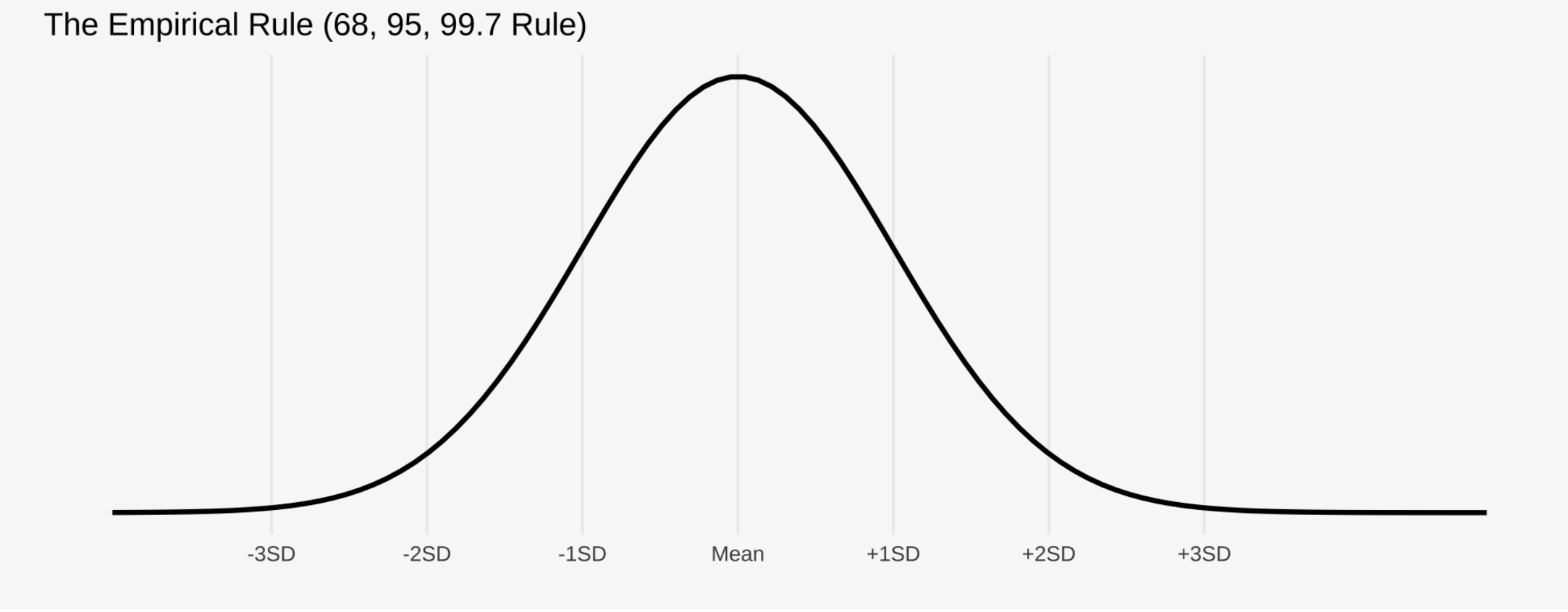


If a variable is approximately normally distributed we can describe its dispersion around the mean using 'standard deviation'.

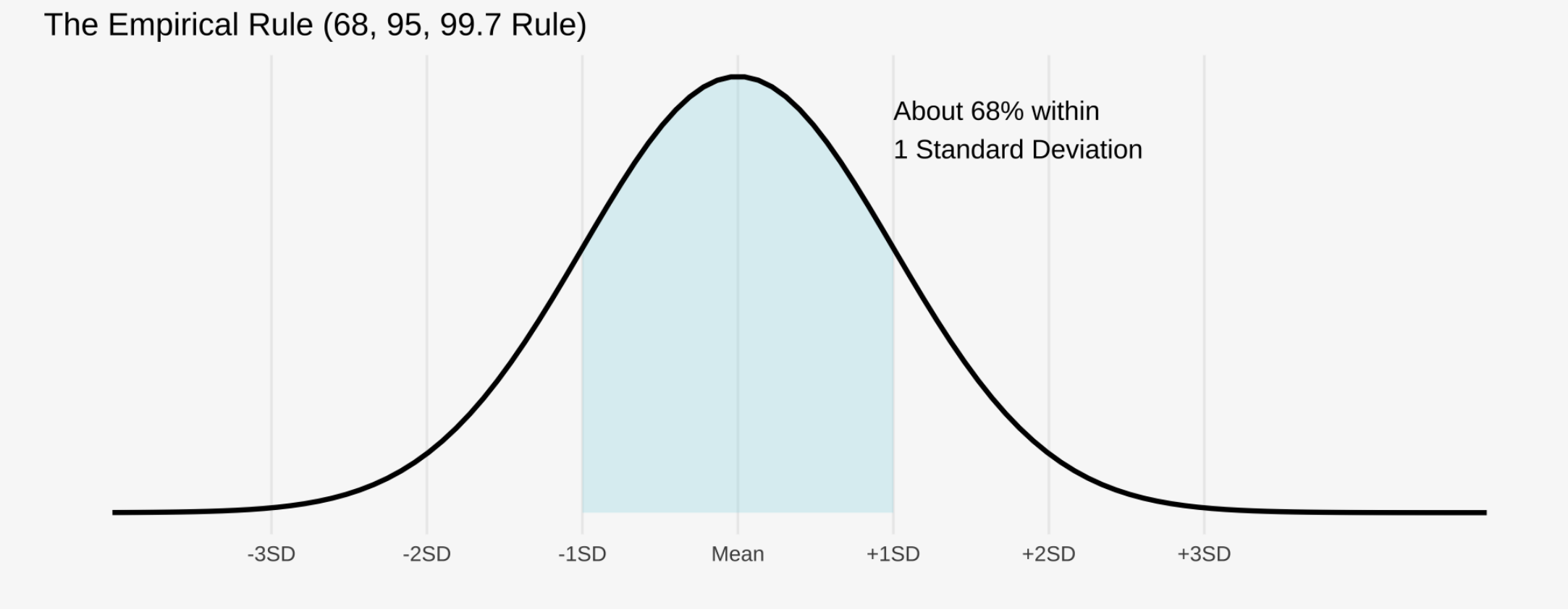
```
sd(idaci$idaci)
```

```
## [1] 5.977357
```

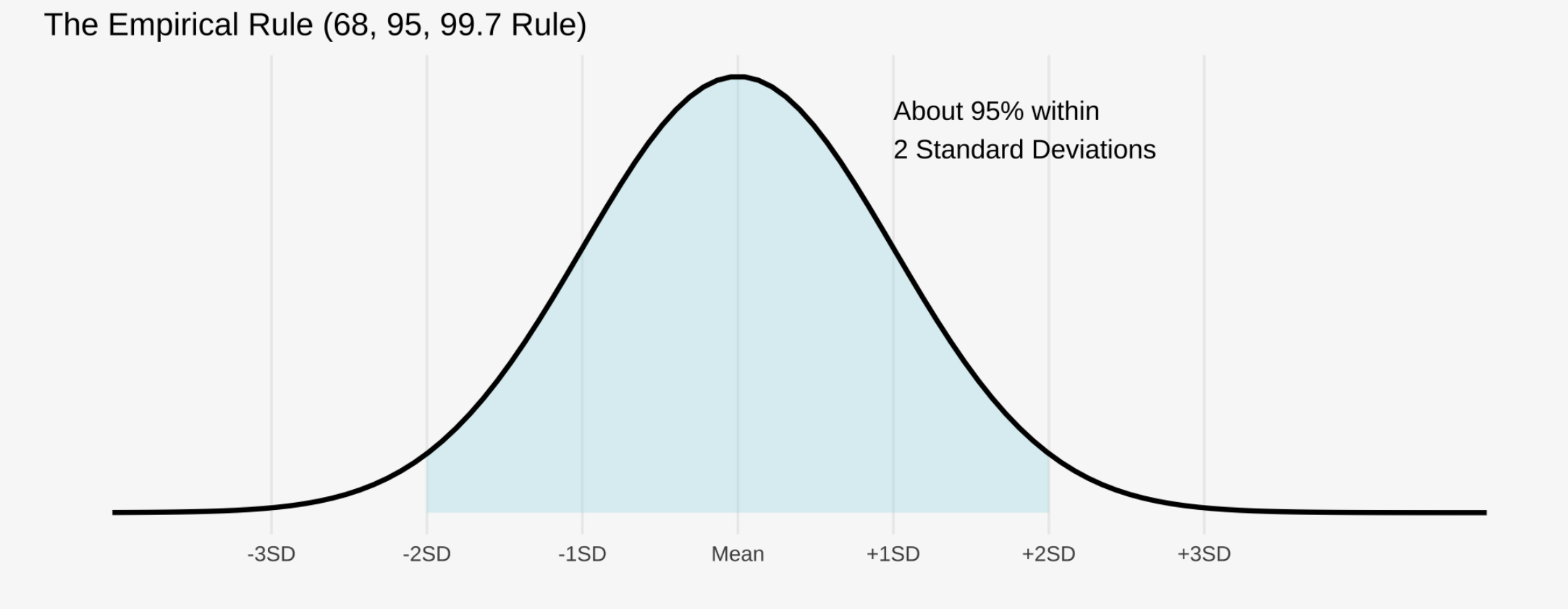
Univariate Descriptive Statistics
Measures of Dispersion



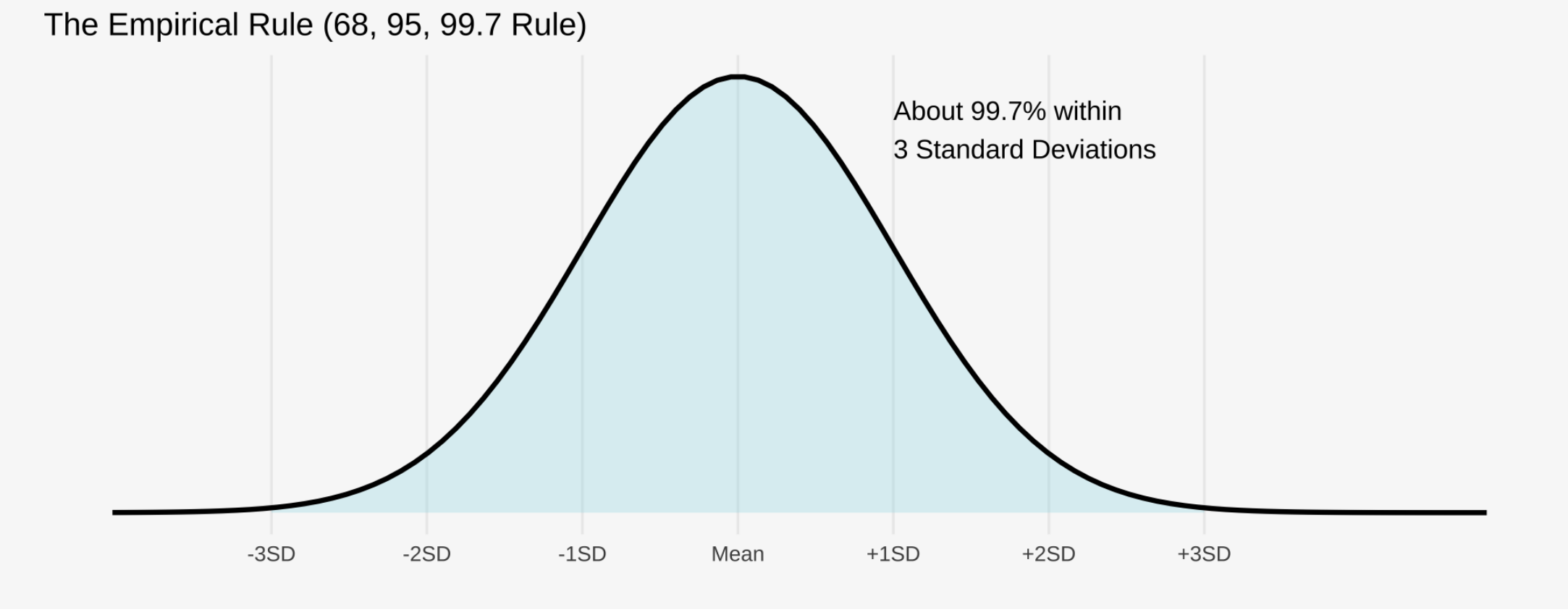
Univariate Descriptive Statistics
Measures of Dispersion



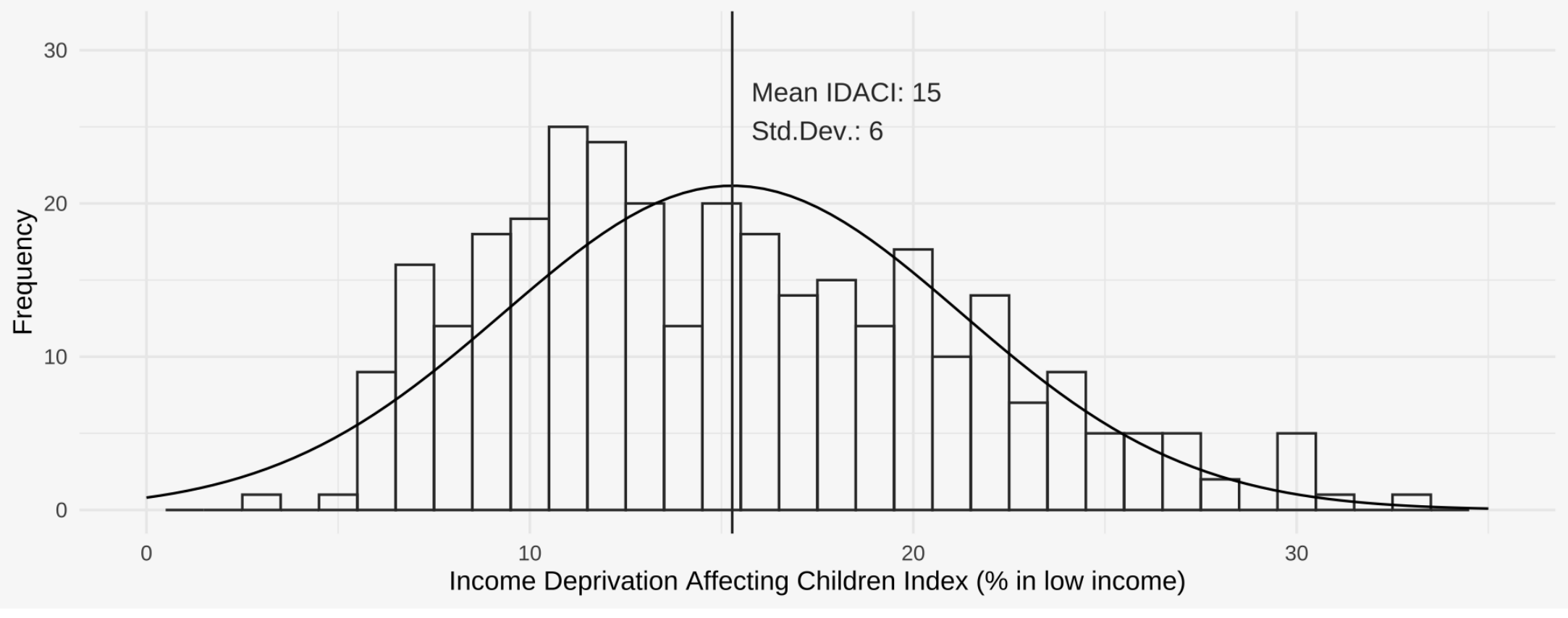
Univariate Descriptive Statistics
Measures of Dispersion



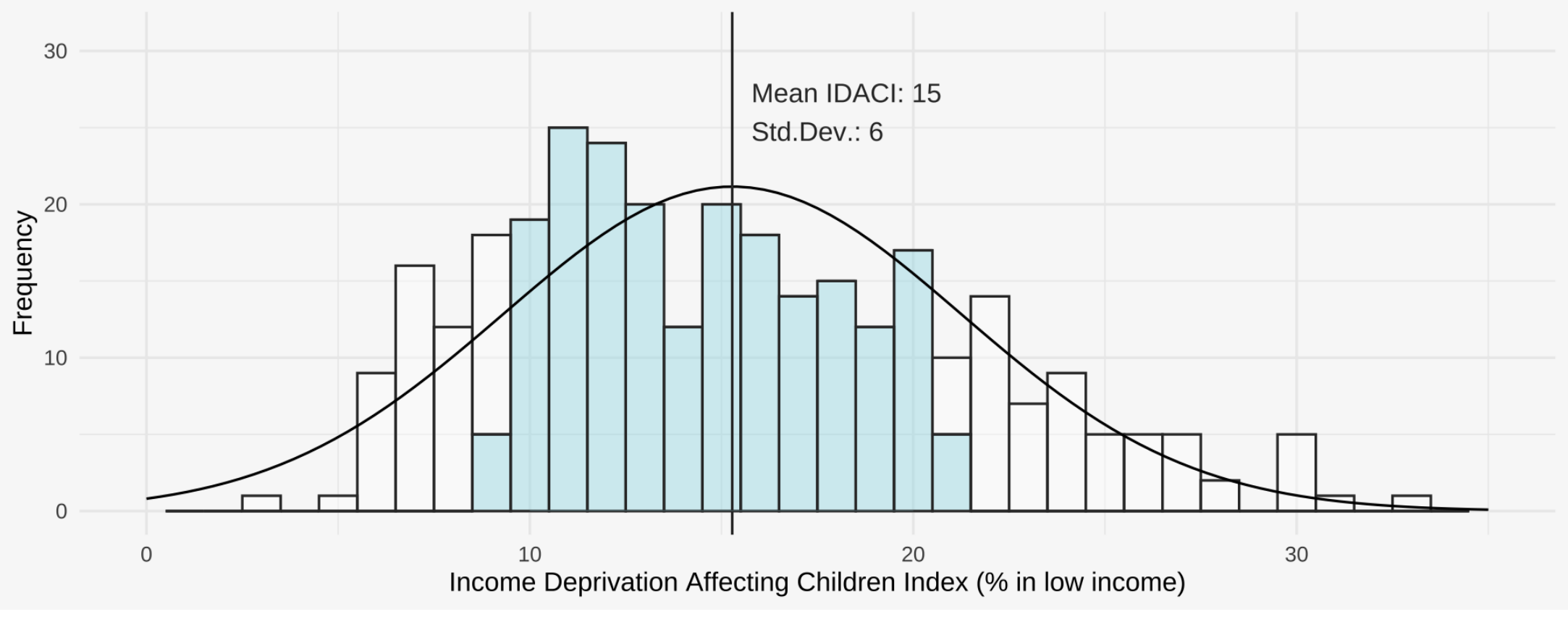
Univariate Descriptive Statistics
Measures of Dispersion



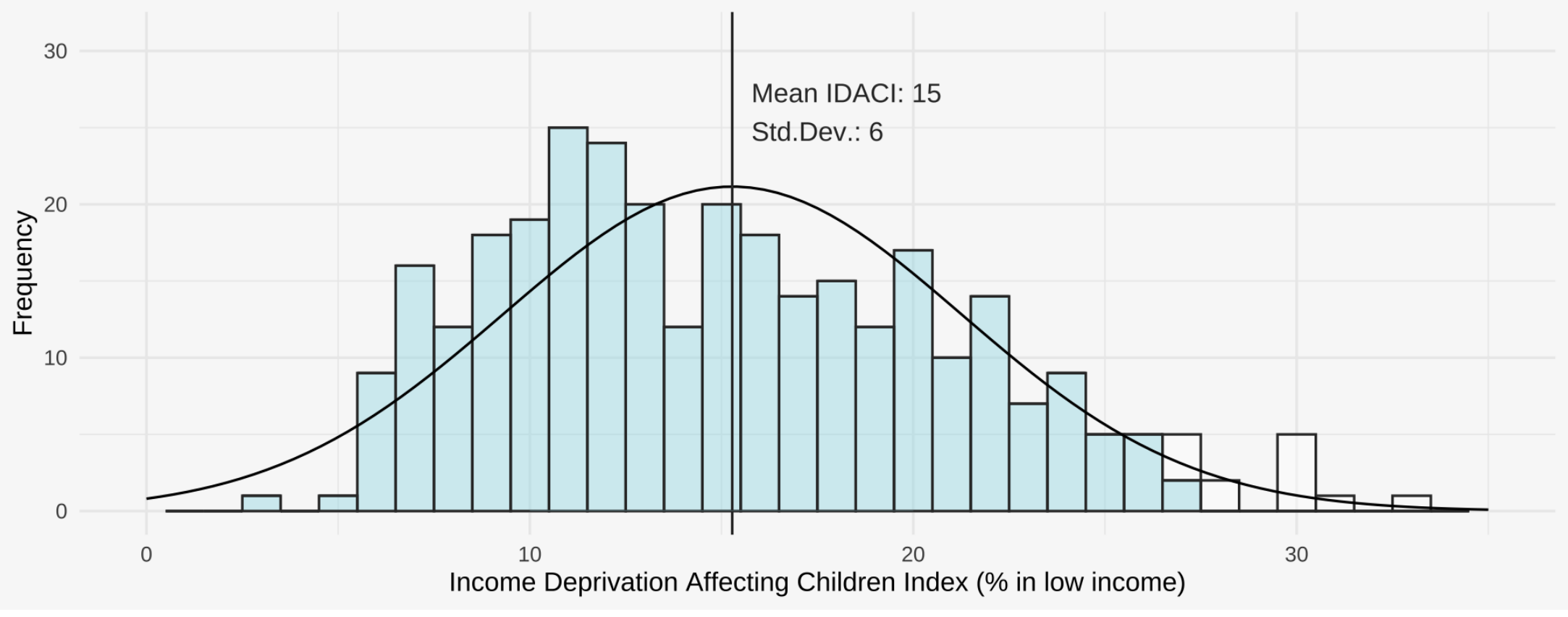
Univariate Descriptive Statistics
Measures of Dispersion



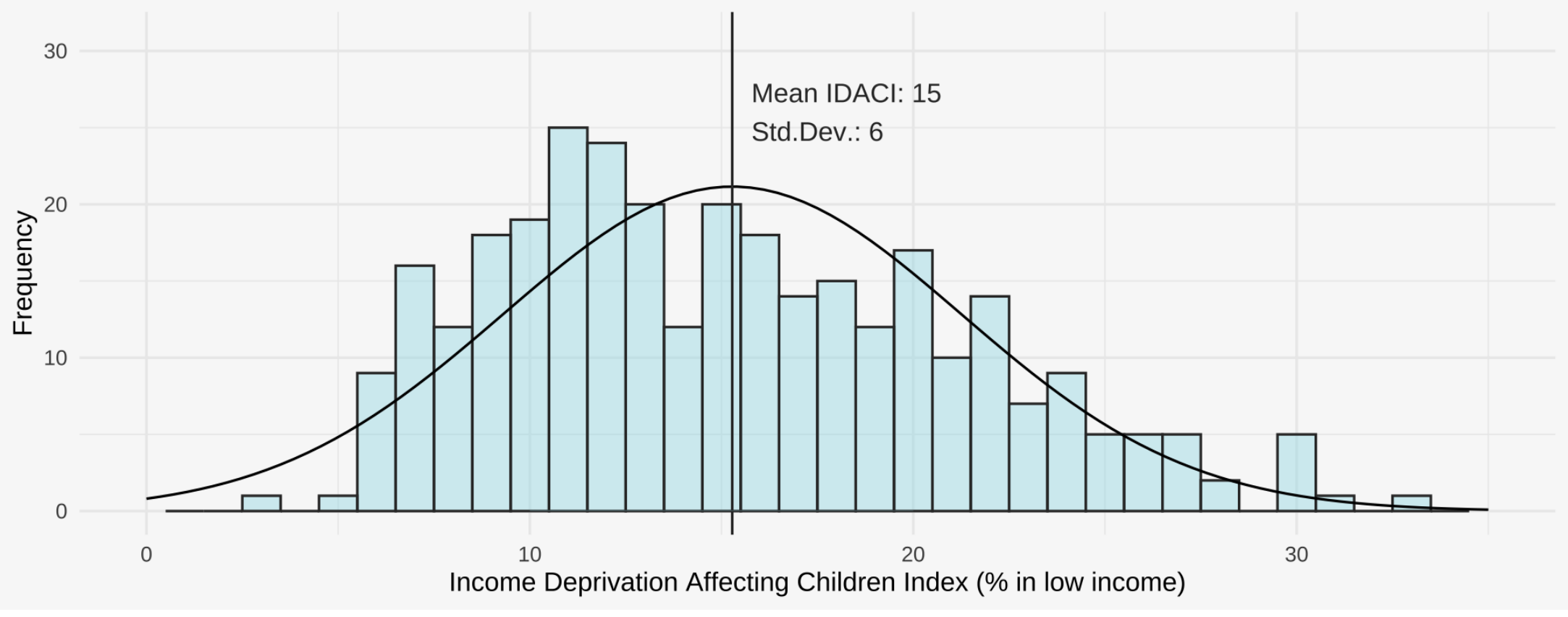
Univariate Descriptive Statistics
Measures of Dispersion



Univariate Descriptive Statistics
Measures of Dispersion



Univariate Descriptive Statistics
Measures of Dispersion



Univariate Descriptive Statistics
Measures of Dispersion

```
# Mean = ~15
# Standard deviation = ~6

idaci %>%
    mutate(
    idaci_1sd = ifelse(idaci > (15-(1*6)) & idaci < (15+(1*6)), TRUE, FALSE),
    idaci_2sd = ifelse(idaci > (15-(2*6)) & idaci < (15+(2*6)), TRUE, FALSE),
    idaci_3sd = ifelse(idaci > (15-(3*6)) & idaci < (15+(3*6)), TRUE, FALSE)
  )
```

```
## # A tibble: 317 × 5
##    lad    idaci idaci_1sd idaci_2sd idaci_3sd
##    <chr>  <dbl> <lgl>     <lgl>     <lgl>
##  1 Bath …  10.4 TRUE      TRUE      TRUE
##  2 Bedfo…  15.1 TRUE      TRUE      TRUE
##  3 Black…  22.8 FALSE     TRUE      TRUE
##  4 Black…  30.7 FALSE     FALSE     TRUE
##  5 Bourn…  14.7 TRUE      TRUE      TRUE
##  6 Brack…   8.9 FALSE     TRUE      TRUE
##  7 Brigh…  15.3 TRUE      TRUE      TRUE
##  8 Brist…  20.6 TRUE      TRUE      TRUE
##  9 Centr…  11.4 TRUE      TRUE      TRUE
## 10 Chesh…  10.7 TRUE      TRUE      TRUE
## # ℹ 307 more rows
```

Univariate Descriptive Statistics
Measures of Dispersion

```
# Mean = ~15
# Standard deviation = ~6

idaci %>%
    mutate(
    idaci_1sd = ifelse(idaci > (15-(1*6)) & idaci < (15+(1*6)), TRUE, FALSE),
    idaci_2sd = ifelse(idaci > (15-(2*6)) & idaci < (15+(2*6)), TRUE, FALSE),
    idaci_3sd = ifelse(idaci > (15-(3*6)) & idaci < (15+(3*6)), TRUE, FALSE)
  ) %>%
  summarise_all(
    ~mean(., na.rm = TRUE)
  )
```

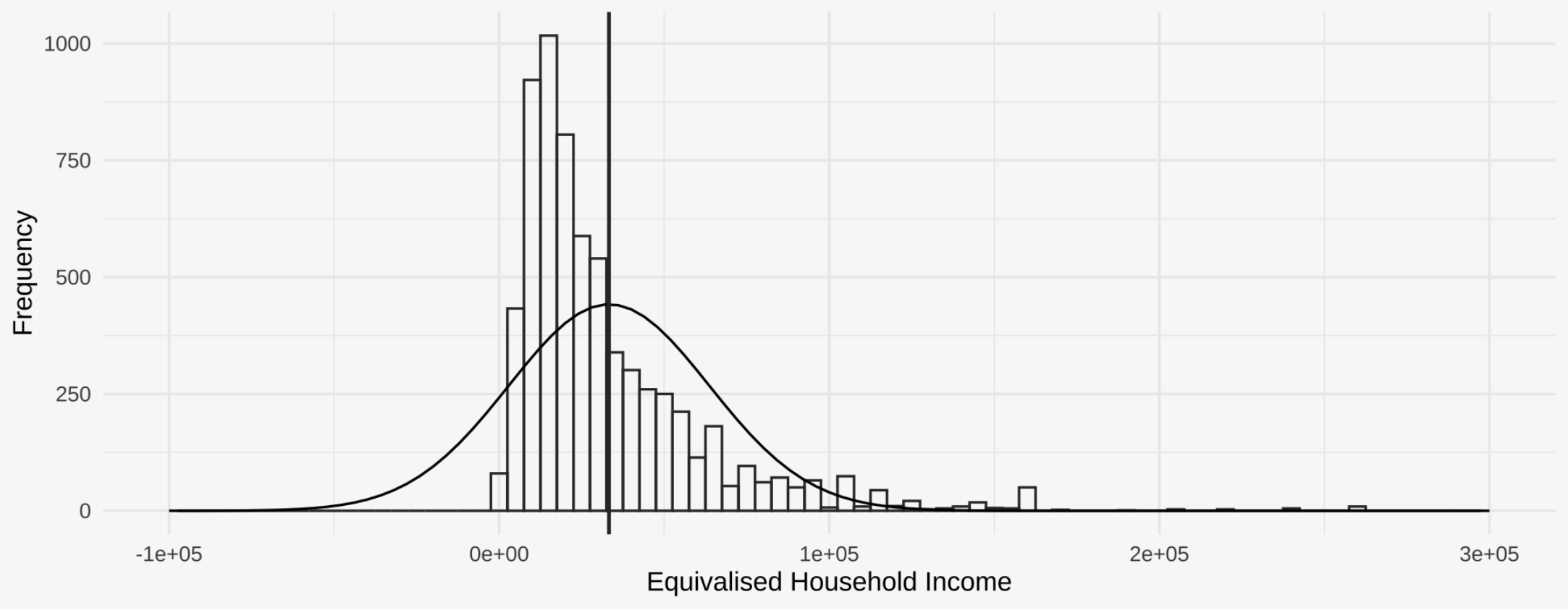
```
## # A tibble: 1 × 5
##     lad idaci idaci_1sd idaci_2sd idaci_3sd
##   <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1    NA  15.3     0.650     0.962         1
```

The Empirical Rule states that within **1 standard deviation** should be approx **68%** of observations, within **2 standard deviations** should be around **95%** of observations, within **3 standard deviations** should be around **99.7%** of observations.

However, we know from the example of income we saw before that sometimes variables are not normally distributed. How can we summarise their distribution?

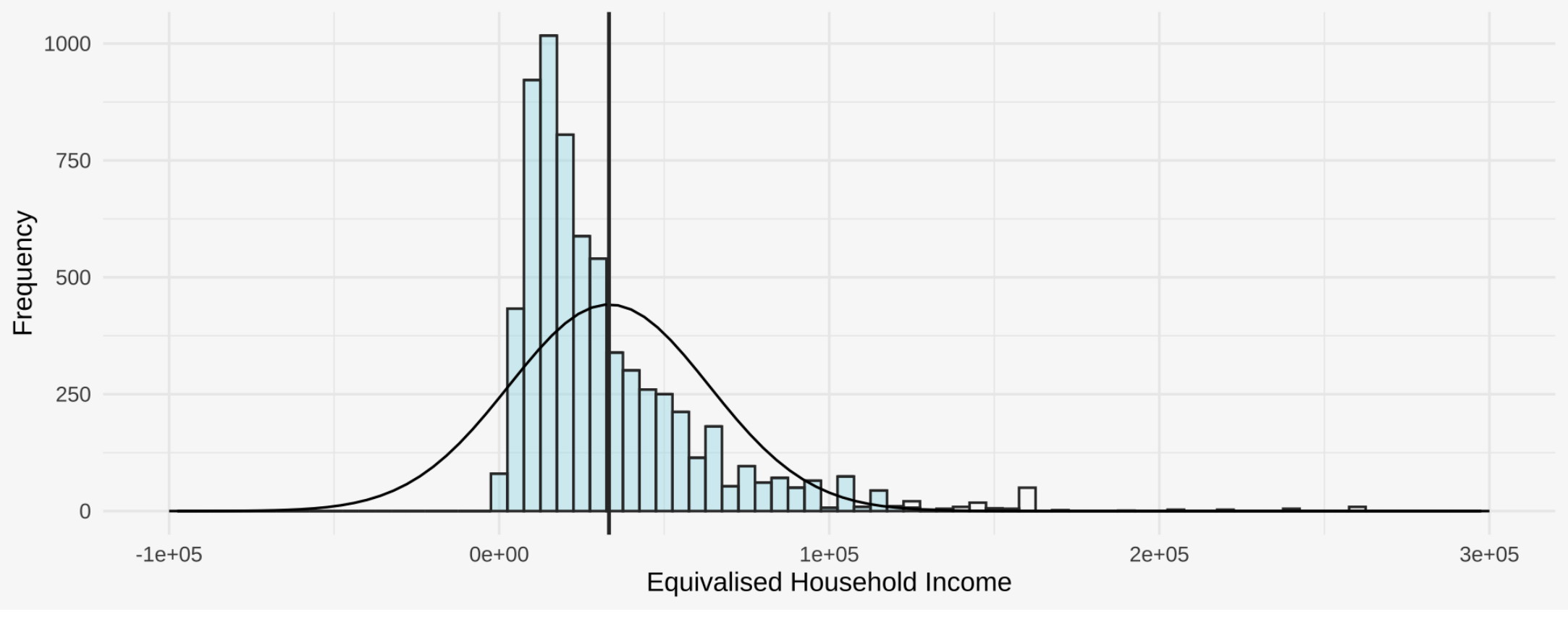
Univariate Descriptive Statistics
Measures of Dispersion: Quantiles

So what happens if we use the empirical rule with non-normally distributed data?



Univariate Descriptive Statistics
Measures of Dispersion: Quantiles

So what happens if we use the empirical rule with non-normally distributed data?



Univariate Descriptive Statistics
Measures of Dispersion: Quantiles

Instead of relying on the Empirical Rule, we can use **quantiles/percentiles** to describe any arbitrary range of our data.

```
# Default quantiles
quantile(hse_cleaned$eqvinc, na.rm = TRUE)
```

```
##          0%         25%         50%
##    271.0843  14300.0000  23442.6230
##         75%        100%
##  43624.1611 262295.0820
```

By default, this gives us a minimum, maximum, and "quartiles". This means, for instance, we can tell that 50% of our data lies between £14,300 (the 25th percentile) and £43,624 (the 75th percentile) because there are 50 percentiles between 25 and 75.

Univariate Descriptive Statistics
Measures of Dispersion: Quantiles

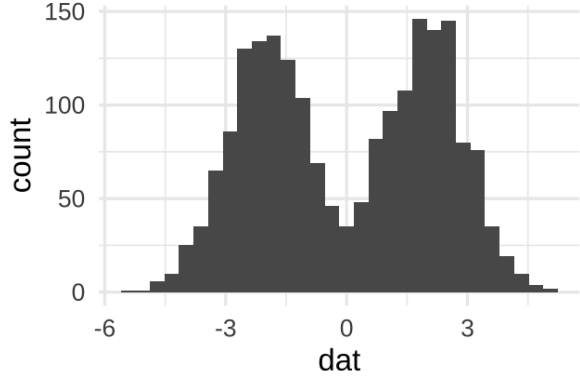
If we wanted a 95% range we would want to take the 2.5th percentile and the 97.5th percentile (as a probability, this would be 0.025 and 0.975). We can do this by editing our quantiles function.

```
# Default quantiles
quantile(hse_cleaned$eqvinc, probs = c(0.025, 0.975), na.rm = TRUE)
```

```
##     2.5%    97.5%
##   4062.5 115000.0
```

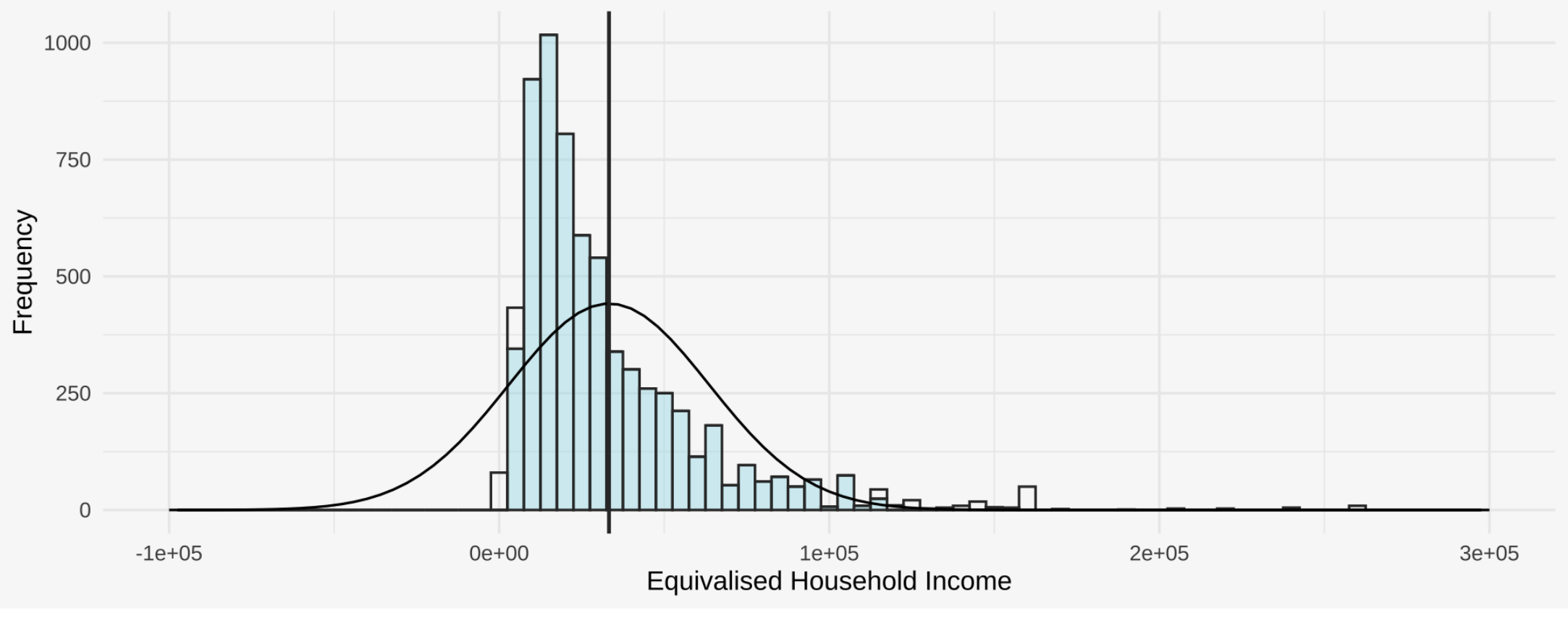
It's useful to provide both quartiles (25th & 75th) and deciles (10th and 90th) to give an idea of a common range and an extreme range.

Quantiles can be used for any kind of continuous data. But you should always plot your continuous data as a histogram because it might have yet stranger features (like bimodal distributions!)



Univariate Descriptive Statistics
Measures of Dispersion: Quantiles

Here are the results if we used quantiles set at pr = 0.025 and pr = 97.5 (95% central distribution of the data)



Variable Type	Central Tendency	Dispersion	Data Vis
Nominal	Percentage of Each Value Frequency Counts janitor::tabyl()	Percentage of Each Value Frequency Counts janitor::tabyl()	Bar Chart geom_bar()/barplot()
Ordinal	Percentage of Each Value Frequency Counts janitor::tabyl() Means/Median/Mode mean(), median(), modeest::mfv()	Percentage of Each Value Frequency Counts janitor::tabyl()	Bar Chart geom_bar()/barplot()
Continuous	Mean/Median/Mode mean(), median(), modeest::mfv()	Standard Deviation + Empirical Rule (if approx normally distributed) sd() Quantiles/Percentiles quantile()	Histogram geom_histogram()/hist()

Week 2 R Exercises

- Today we will be using a different kind of tutorial using an Rmarkdown document. Check the files below the lecture slides in Week 2 under the learning materials menu.
- **Follow the worksheet** to learn how to calculate univariate descriptive statistics and visualise single variables in **R** using a few different methods.
- If you finish the tutorial with a lot of time remaining, **try out the Week 2 challenge**! Make sure you finish this workbook and the challenge, as well as your required reading in your own independent study time!