Online CDS 101 *Introduction to Computational and Data Sciences*

Data distributions

Cumulative distribution functions

Download and load the dataset

You can follow along by downloading and loading the dataset by placing the following *setup* code block at the top of a R Markdown file.

````
```{r setup, include = FALSE}
# Load required packages
library(tidyverse)
# Load datasets
county <- read_rds(url("http://data.cds101.com/county_complete.rds"))
```
````

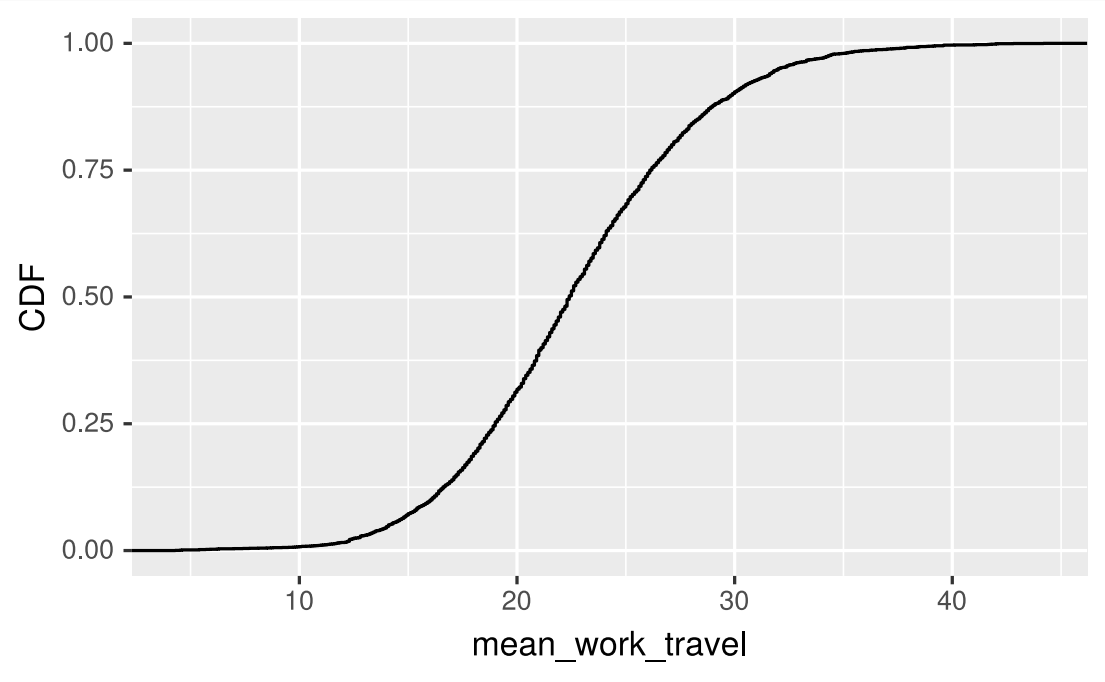
Data by percentile rank

- Probability mass functions (PMFs) are handy exploratory tools, but as with histograms, the bin width can strongly influence what your plot looks like
- The density plots offer a reasonable alternative to the PMFs, however, comparing distributions with different widths, multiple outliers, or very skewed data can still be challenging.
- A convenient way to overcome this problem if we convert the data into a sorted list of percentile ranks
- **Advantages**
    - Don't need to select a bin width
    - Easier to compare similarities and differences of different data distributions
    - Different classes of data distributions have distinct shapes
- The **cumulative distribution function** (CDF) lets us map between percentile rank and each value in a data column

Creating CDFs in R

`ggplot2` comes with a handy convenience function `stat_ecdf()`, which lets you create and visualize CDFs

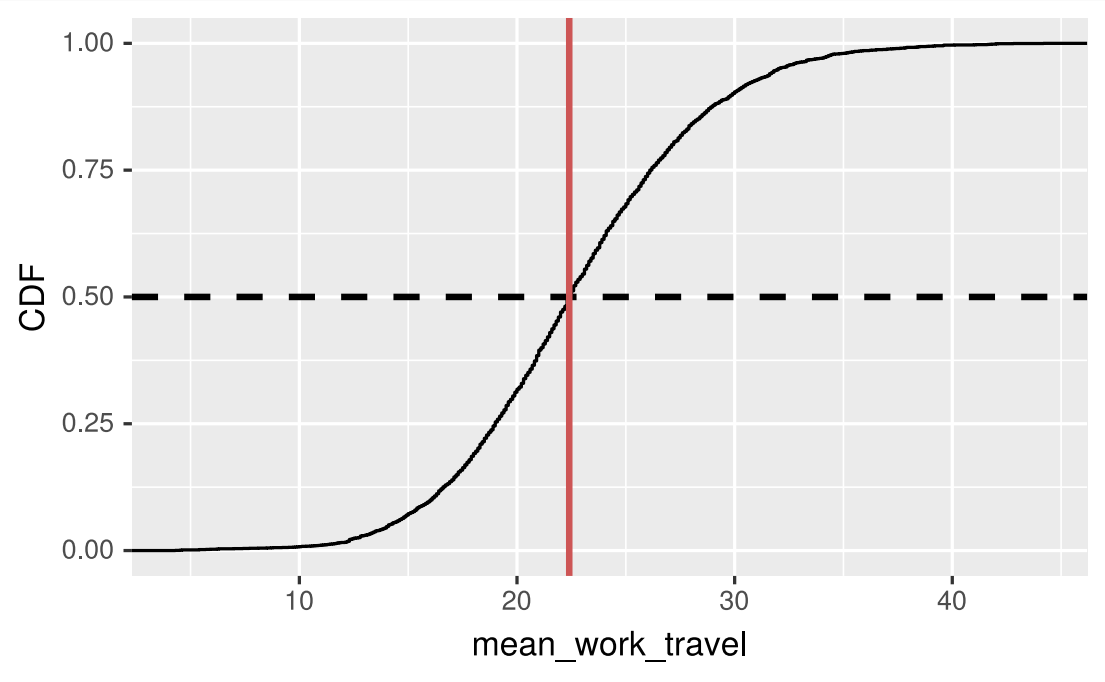
```
ggplot(county) +
  stat_ecdf(mapping = aes(x = mean_work_travel)) +
  labs(y = "CDF")
```



Creating CDFs in R

`ggplot2` comes with a handy convenience function `stat_ecdf()`, which lets you create and visualize CDFs

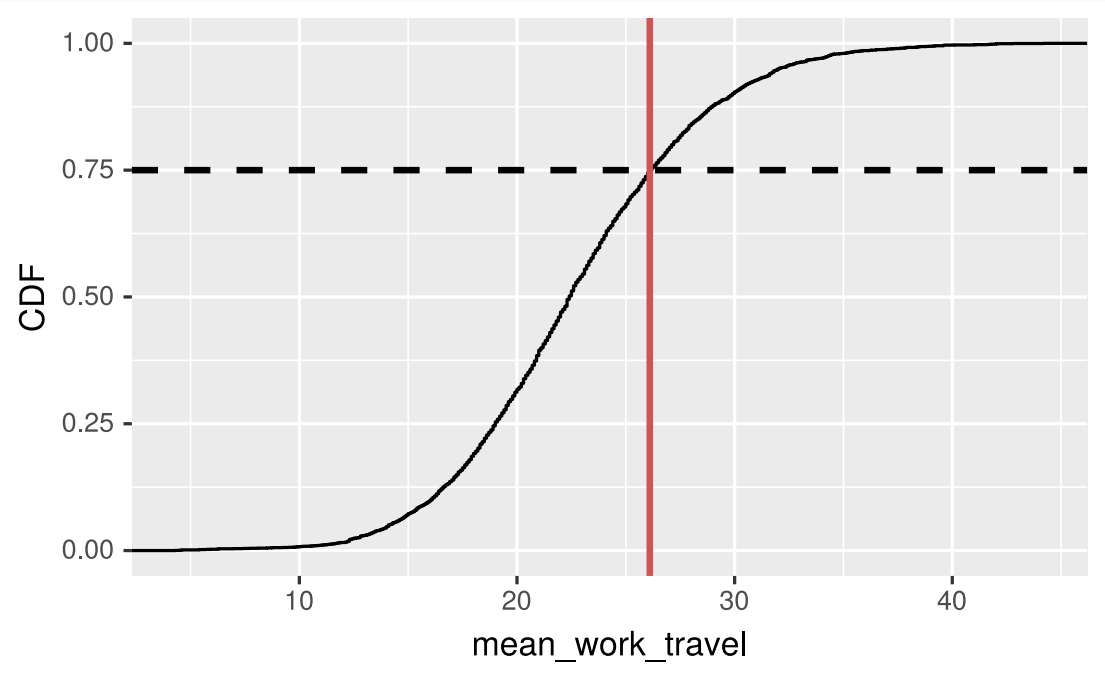
```
ggplot(county) +
  stat_ecdf(mapping = aes(x = mean_work_travel)) +
  labs(y = "CDF")
```



Creating CDFs in R

`ggplot2` comes with a handy convenience function `stat_ecdf()`, which lets you create and visualize CDFs

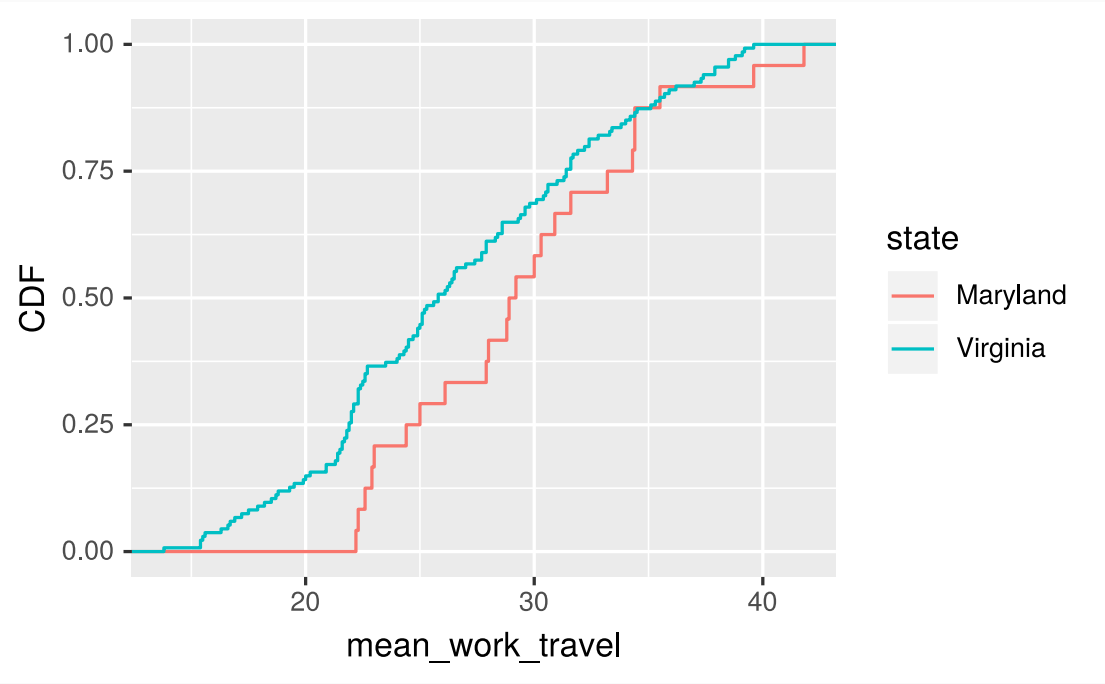
```
ggplot(county) +
  stat_ecdf(mapping = aes(x = mean_work_travel)) +
  labs(y = "CDF")
```



Creating CDFs in R

We can do all the usual operations, such as grouping by state

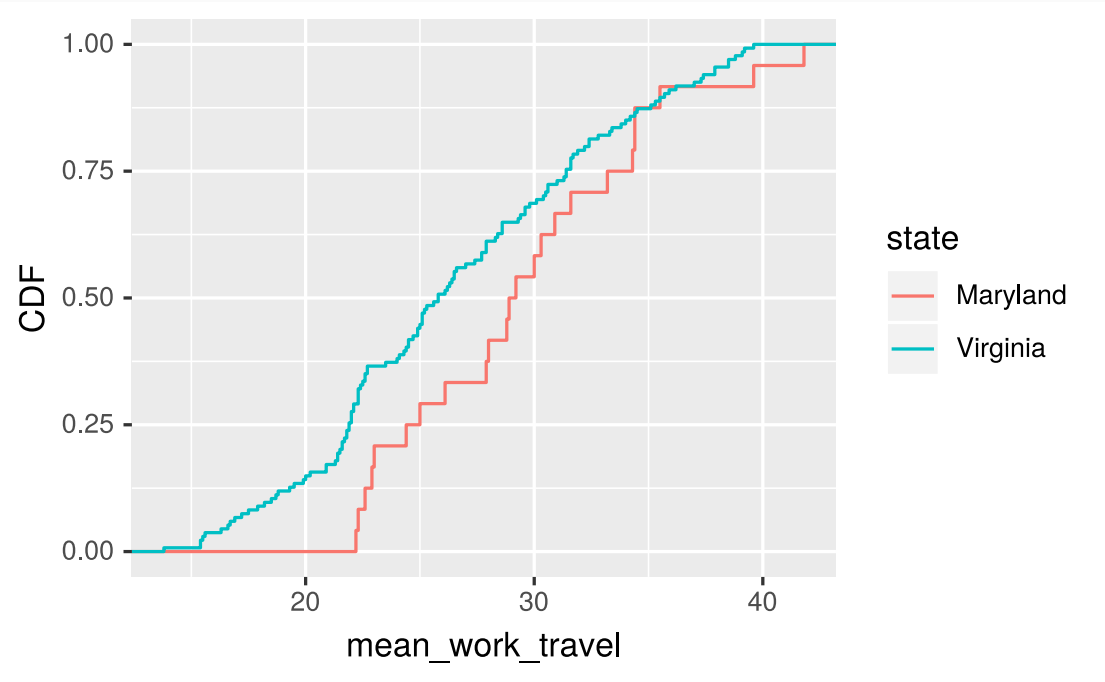
```
county %>%
  filter(state == "Virginia" | state == "Maryland") %>%
  ggplot() +
  stat_ecdf(mapping = aes(x = mean_work_travel, color = state)) +
  labs(y = "CDF")
```



Creating CDFs in R

We can do all the usual operations, such as grouping by state

```
county %>%
  filter(state == "Virginia" | state == "Maryland") %>%
  ggplot() +
  stat_ecdf(mapping = aes(x = mean_work_travel, color = state)) +
  labs(y = "CDF")
```



Computing the CDF

To compute the CDF, we use the `cume_dist()` function along with `filter()`, `group_by()`, and `mutate()`:

```
va_md_cdf_df <- county %>%
  filter(state == "Virginia" | state == "Maryland") %>%
  group_by(state) %>%
  mutate(cdf = cume_dist(mean_work_travel)) %>%
  select(state, mean_work_travel, cdf)
```

Get CDF data out of plot

state	mean_work_travel	cdf
Virginia	13.8	0.0074627
Virginia	15.4	0.0223881
Virginia	15.4	0.0223881
Virginia	15.5	0.0298507
Virginia	15.6	0.0373134
Virginia	16.3	0.0447761
Virginia	16.6	0.0522388
Virginia	16.7	0.0597015
Virginia	16.9	0.0671642
Virginia	17.2	0.0746269

Credits

License

Creative Commons Attribution-NonCommerical-ShareAlike 4.0 International