

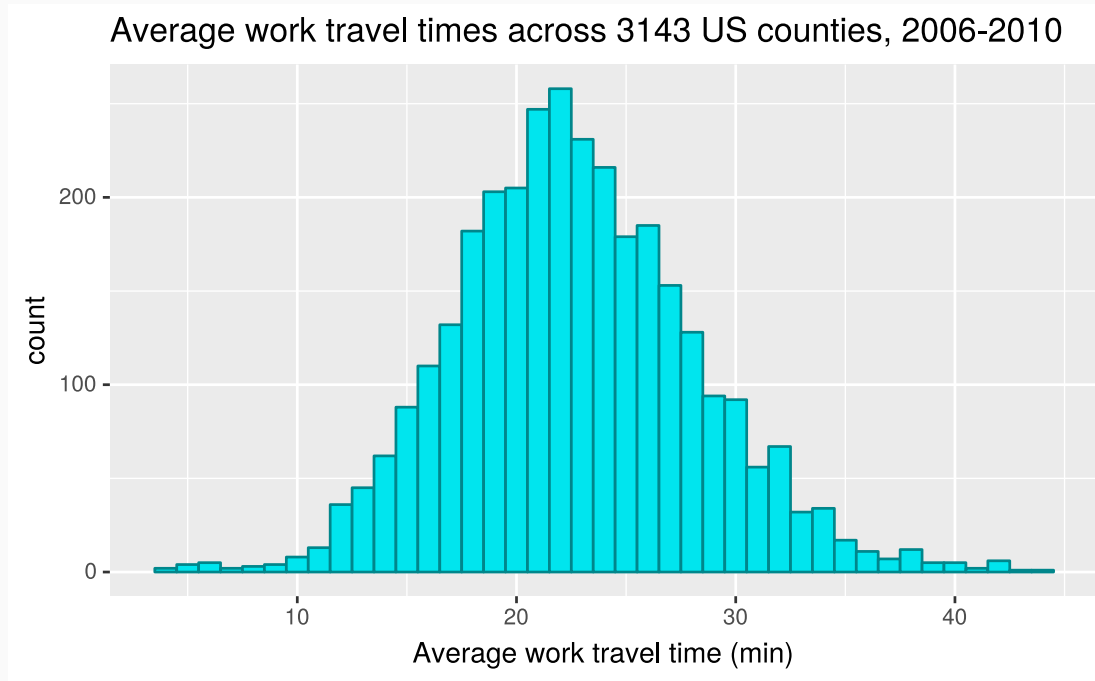
Data distributions

Quantifying data distributions in R



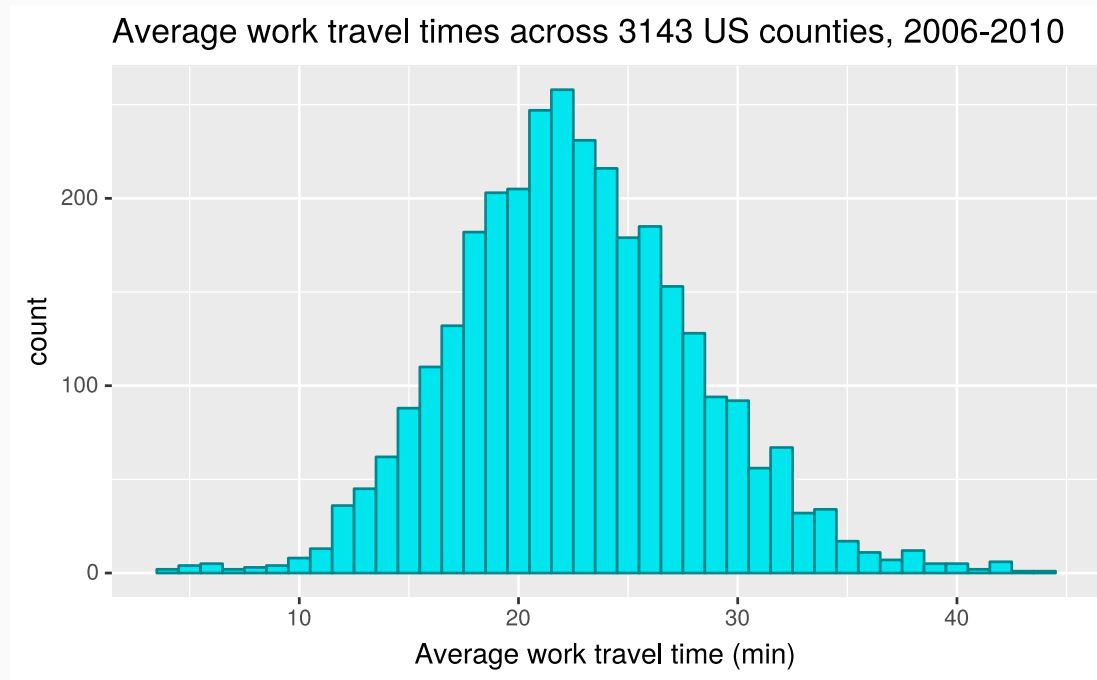
Example data distribution

The following distribution comes from data posted by the US Census Bureau:



Example data distribution

The following distribution comes from data posted by the US Census Bureau:



How can we quantify the shape of this distribution?

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"))
```
```

Useful statistical functions

The following R functions will be useful for computing basic statistical measures of any numerical data column (variable)

- `mean()`: Computes the average
- `median()`: Computes the median
- `min()`: Finds the minimum value
- `max()`: Finds the maximum value
- `sd()`: Computes the standard deviation
- `IQR()`: Computes the interquartile range
- `percent_rank()`: Computes percentiles

Using the statistical functions

Every function except `percent_rank()` will always return a single quantity

The `summarize()` function is appropriate here:

```
county %>%  
  summarize(  
    mean = mean(mean_work_travel),  
    median = median(mean_work_travel),  
    min = min(mean_work_travel),  
    max = max(mean_work_travel),  
    sd = sd(mean_work_travel),  
    iqr = IQR(mean_work_travel)  
  )
```

| mean | median | min | max | sd | iqr |
|----------|--------|-----|------|----------|-----|
| 22.72558 | 22.4 | 4.3 | 44.2 | 5.514159 | 7.1 |

Using the statistical functions

`percent_rank()` operates on the full column of values, so it needs to be paired with `mutate()`

Once we have the percentiles, we can find the cutoff value for each percentile

```
county %>%
  mutate(
    percentile = percent_rank(mean_work_travel),
    quartile = case_when(
      percentile < 0.25 ~ "Q1",
      between(percentile, 0.25, 0.50) ~ "Q2",
      between(percentile, 0.50, 0.75) ~ "Q3",
      percentile >= 0.75 ~ "Q4"
    )
  ) %>%
  group_by(quartile) %>%
  summarize(cutoff = max(mean_work_travel))
```

| Q1 | Q2 | Q3 | Q4 |
|----|------|------|------|
| 19 | 22.4 | 26.1 | 44.2 |

Using the statistical functions

`percent_rank()` operates on the full column of values, so it needs to be paired with `mutate()`

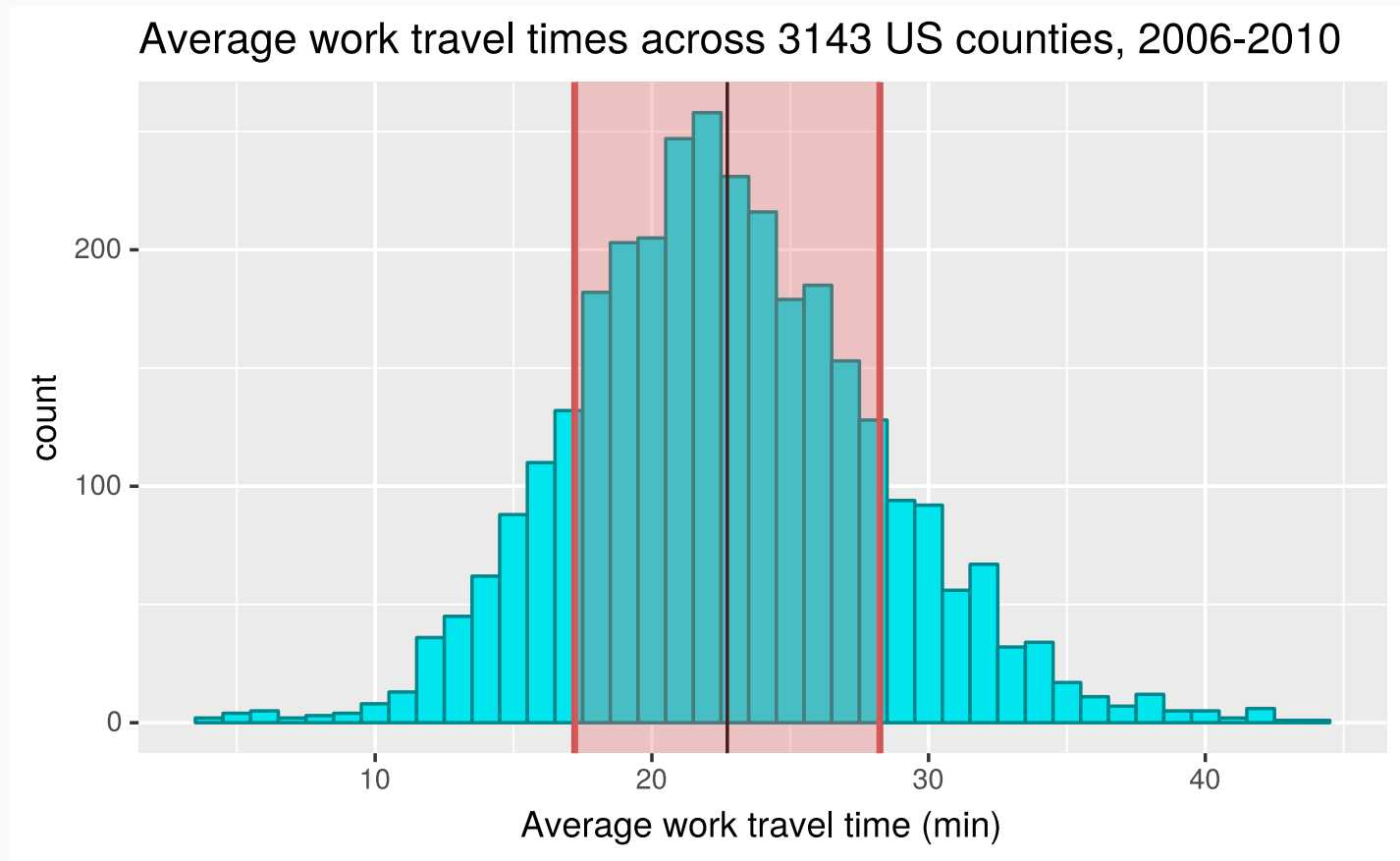
Once we have the percentiles, we can find the cutoff value for each percentile

```
county %>%  
  mutate(  
    percentile = percent_rank(mean_work_travel),  
    quartile = case_when(  
      percentile < 0.25 ~ "Q1", # case_when() similar to if_else()  
      between(percentile, 0.25, 0.50) ~ "Q2", # label between 0 and 0.25 as Q1,  
      between(percentile, 0.50, 0.75) ~ "Q3", # between 0.25 and 0.50 as Q2,  
      percentile >= 0.75 ~ "Q4" # between 0.50 and 0.75 as Q3,  
      # and 0.75 to 1.00 as Q4  
    )  
  ) %>%  
  group_by(quartile) %>%  
  summarize(cutoff = max(mean_work_travel)) # cutoff is maximum in quartile
```

| Q1 | Q2 | Q3 | Q4 |
|----|------|------|------|
| 19 | 22.4 | 26.1 | 44.2 |

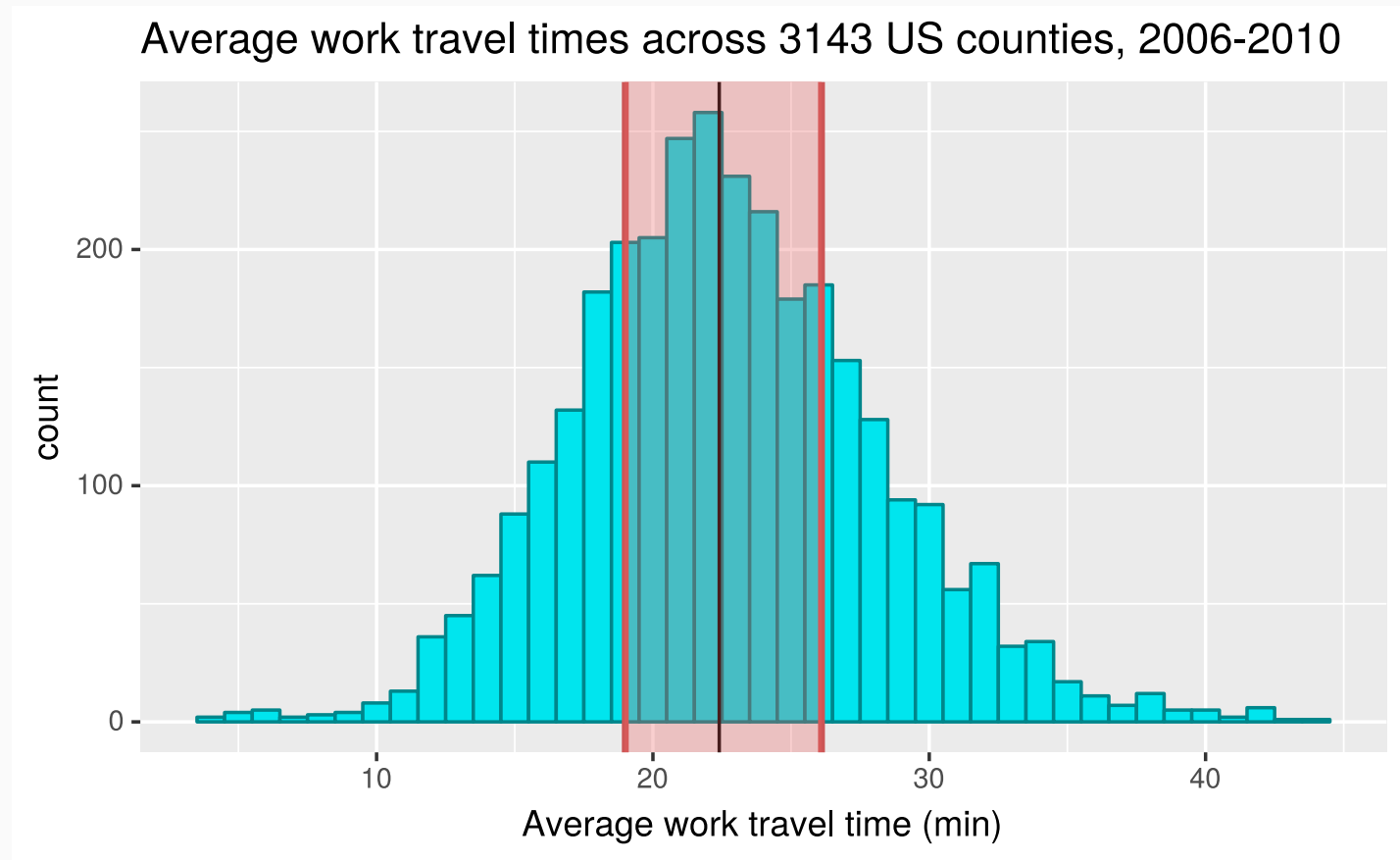
Interpreting summary statistics: mean, sd

One standard deviation above and below the mean



Interpreting summary statistics: median, IQR

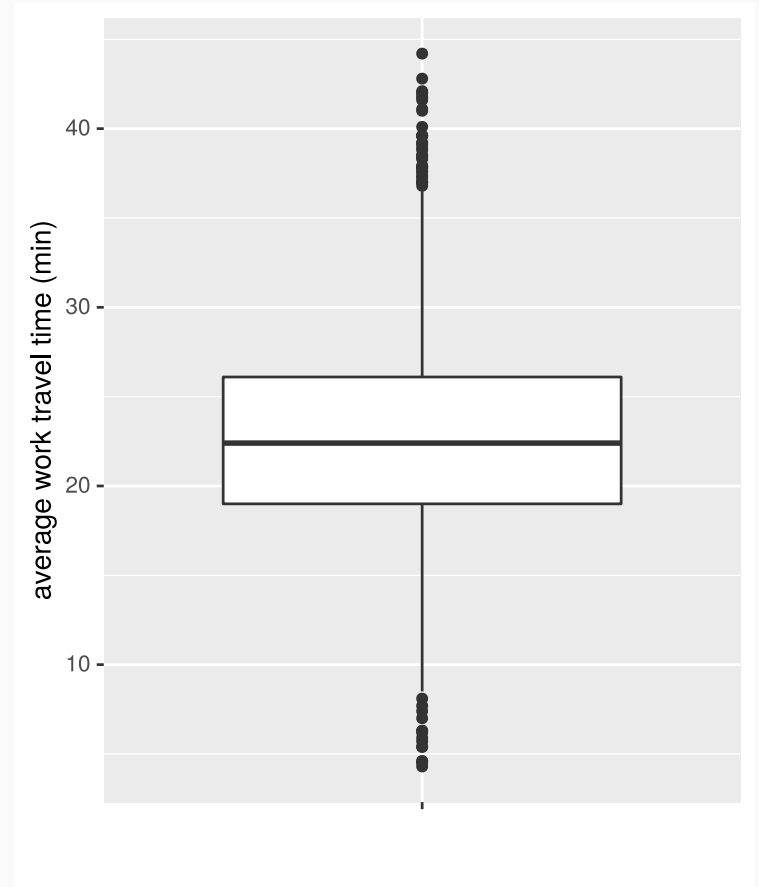
The median and inter-quartile range



Boxplot of the data

To create a boxplot for this dataset:

```
ggplot(county) +  
  geom_boxplot(  
    mapping = aes(  
      x = "",  
      y = mean_work_travel  
    )  
  ) +  
  xlab("") +  
  ylab("average work travel time (min)")
```

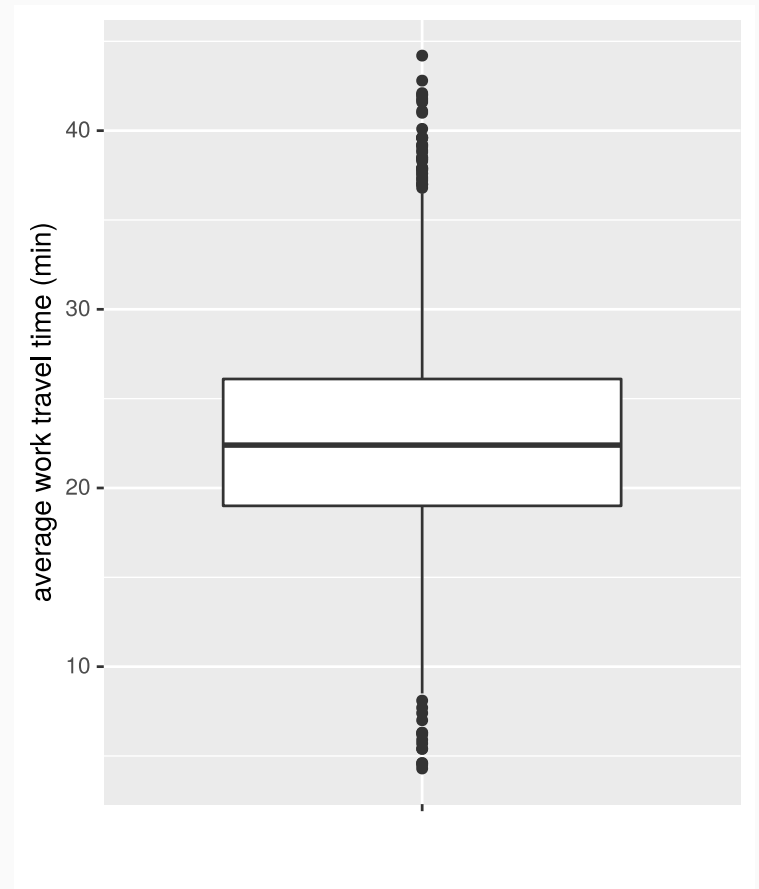


Boxplot of the data

To create a boxplot for this dataset:

```
ggplot(county) +  
  geom_boxplot(  
    mapping = aes(  
      x = "",  
      y = mean_work_travel  
    )  
  ) +  
  xlab("") +  
  ylab("average work travel time (min)")
```

- The `aes` function in `geom_boxplot` requires both `x` and `y` for inputs
- `x` should be a categorical variable, `y` should be a numerical variable
- If you only want to plot a single boxplot, then set `x = ""`



Credits

License

Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International