Informatics 143

Information Visualization

Lecture 6

These course materials are based on books from Claus O. Wilke, Kieran Healy, Edward R. Tufte, Alberto Cairo, Colin Ware, Tamara Munzner, and others.
Powerpoint theme by Prof. André van der Hoek.

Visualization of Distributions

- Usefull to visualize how a set of numbers is distributed
  - E.g. how many students of a certain age are enrolled in ICS

  - Very informal definition of distribution: *The relative proportions of the diferent numbers is called "the distribution" of these numbers.*

Visualization of Distributions

- Usefull to visualize how a set of numbers is distributed
  - E.g. how many students of a certain age are enrolled in ICS
- Data has:
  - At least one set of values (qualitative or quantitative)
- *Some* standard geometrical mappings:
  - Histograms plot
  - Density plots
  - **Cumulative densities**
  - **Boxplots, violins**

Visualization of Distributions

- One problem with histograms and density plots:
  - Both depend on user selected parameters
  - They result from an *interpretation* of the data

Visualization of Distributions

- One problem with histograms and density plots:
  - Both depend on user selected parameters
  - They result from an *interpretation* of the data

- Alternative (bad):
  - Just visualize every data point as a large point cloud
  - Not feasible for large datasets
  - Too much emphasis on individual data points

Visualization of Distributions

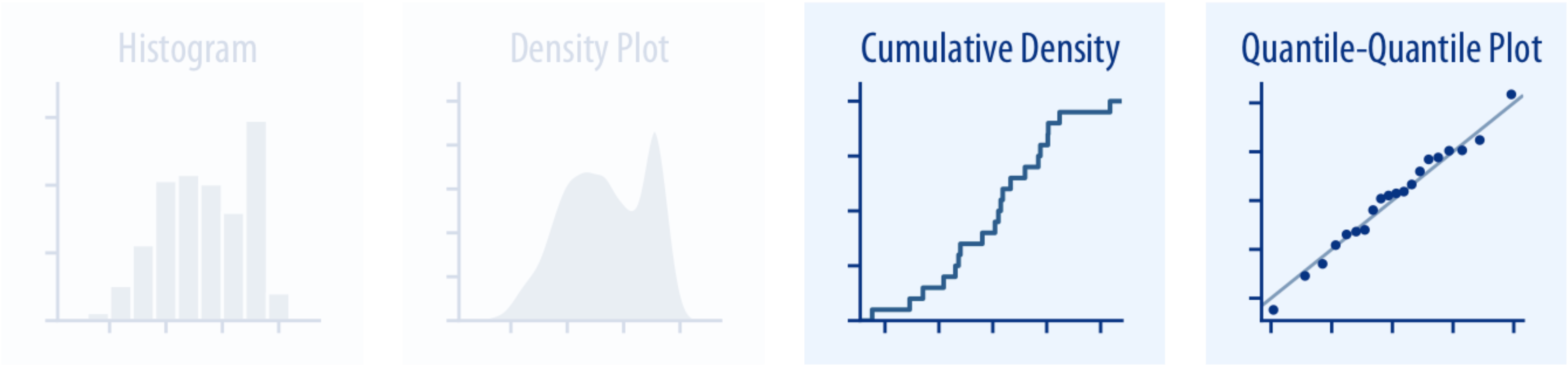
- One problem with histograms and density plots:
  - Both depend on user selected parameters
  - They result from an *interpretation* of the data

- Alternative (bad):
  - Just visualize every data point as a large point cloud
  - Not feasible for large datasets
  - Too much emphasis on individual data points

- Better alternatives:
  - Empirical cumulative distribution functions (ECDFs)
  - Quantile-quantile plots

Visualization of Distributions

- More faithfull data representations



- No arbitrary parameter choices
- All the data is displayed
- Drawback : less intuitive, thus more difficult to interpret

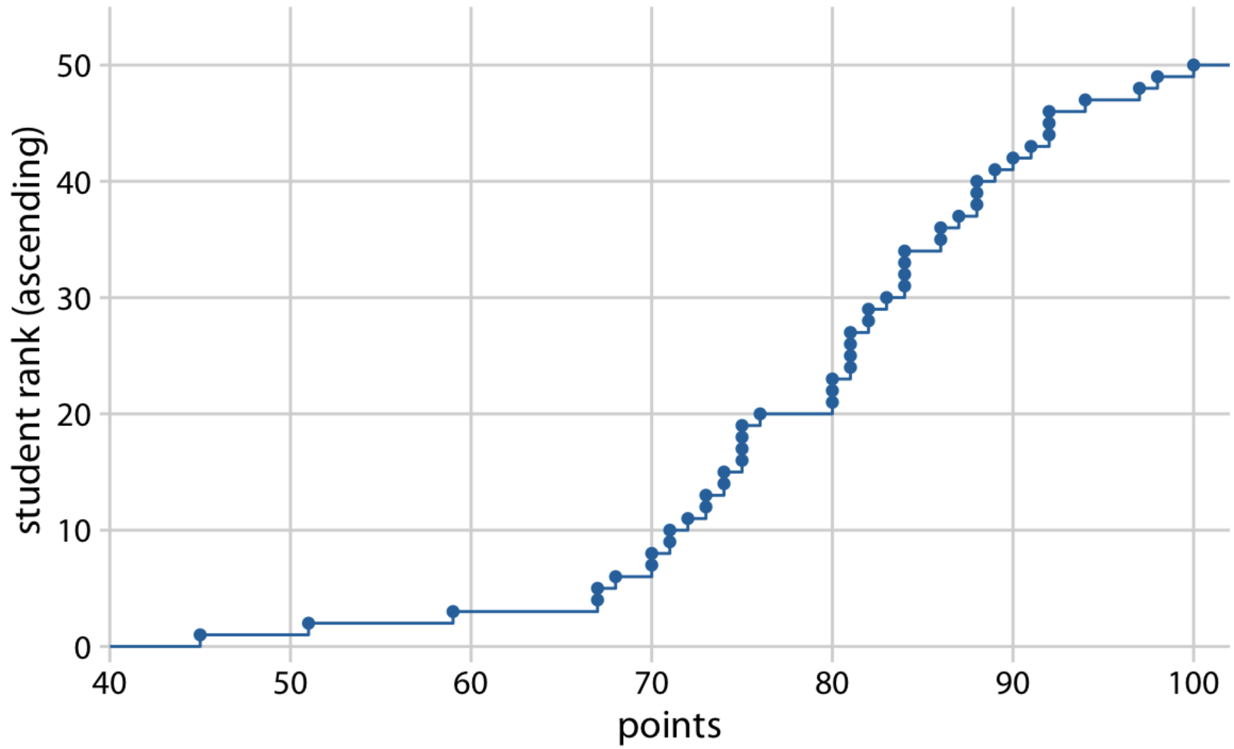
Visualization of Distributions: ECDFs

- Example: dataset of student grades

```
# setting a seed for a random number generator
set.seed(31415)

# creating the random grades
points = round(c(rnorm(47, mean = 87, sd = 10), 45, 51, 67))

# grades cannot be greater than 100...
points[points > 100] <- 100

# organize into a dataframe and rank
student_data <- data.frame(points,
                           rank = rank(points, ties.method = "random"))
```

Visualization of Distributions: ECDFs

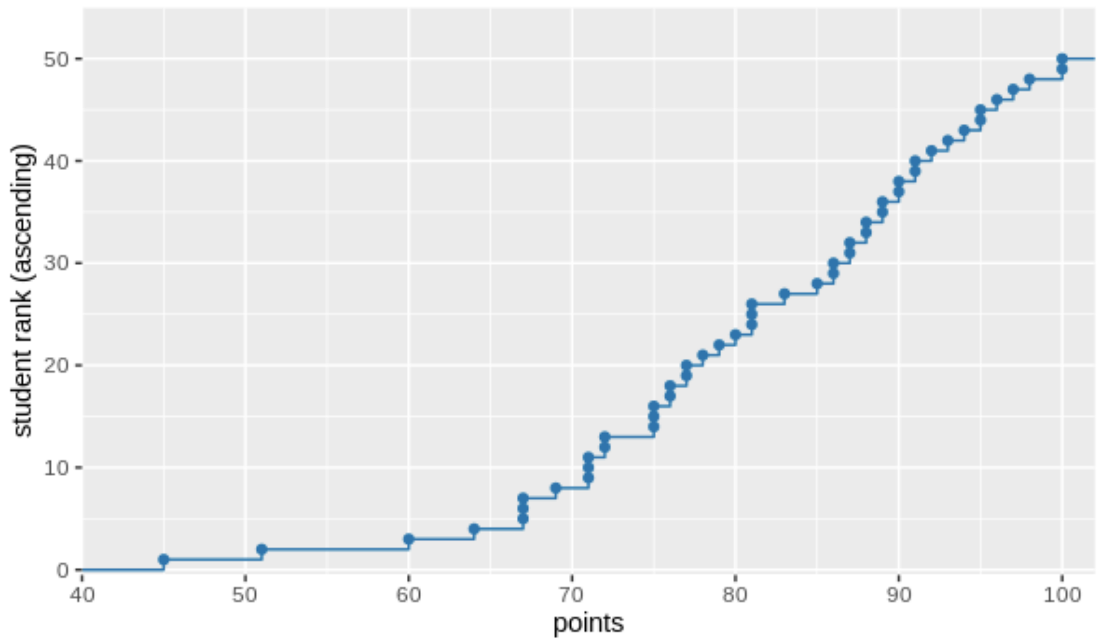
- Example: ECDF of the dataset of student grades



Visualization of Distributions: ECDFs

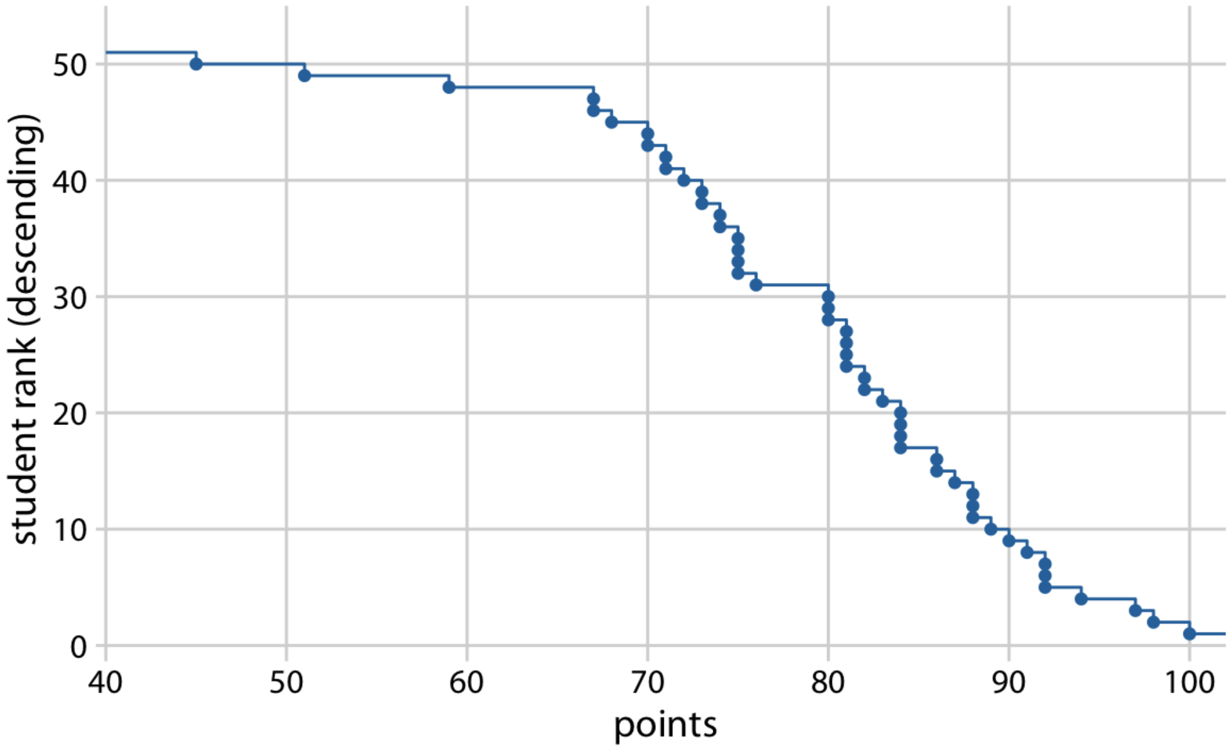
- How to create in ggplot2?
  - stat_ecdf()

```
ggplot(student_data, aes(x = points, y = 50*..y..)) +
    stat_ecdf(geom = "step", color = "#0072B2") +
    geom_point(aes(y = rank), color = "#0072B2") +
    scale_x_continuous(limits = c(40, 102), expand = c(0, 0),
                       breaks = 10*(4:10)) +
    scale_y_continuous(limits = c(-.5, 55), expand = c(0, 0),
                       name = "student rank (ascending)")
```



Visualization of Distributions: ECDFs

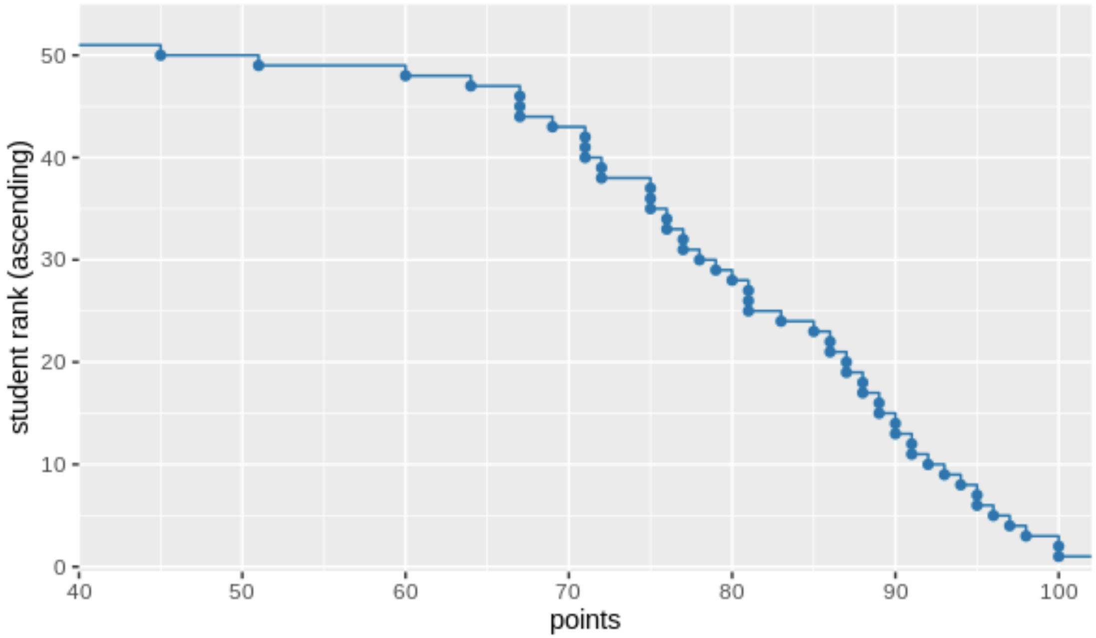
- Example: ECDF of the dataset of student grades in the *descending* order



Visualization of Distributions: ECDFs

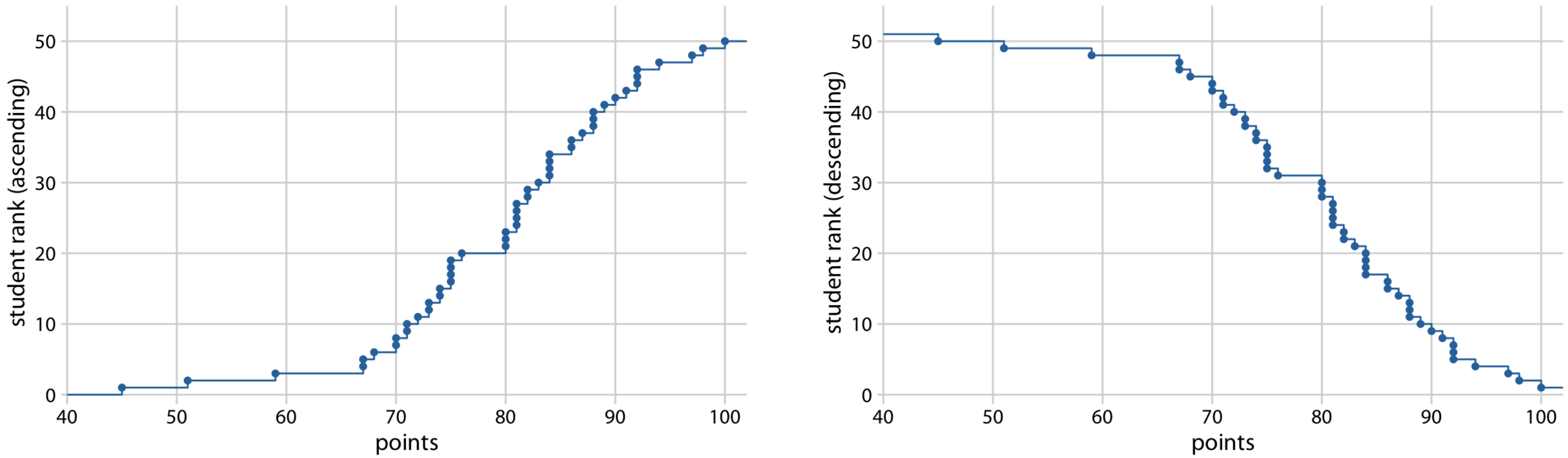
- How to create a descending ECDF in ggplot2?
  - stat_ecdf()

```
ggplot(student_data, aes(x = points, y = 51-50*..y..)) +
    stat_ecdf(geom = "step", color = "#0072B2") +
    geom_point(aes(y = 51-rank), color = "#0072B2") +
    scale_x_continuous(limits = c(40, 102), expand = c(0, 0),
                       breaks = 10*(4:10)) +
    scale_y_continuous(limits = c(-.5, 55), expand = c(0, 0),
                       name = "student rank (ascending)")
```



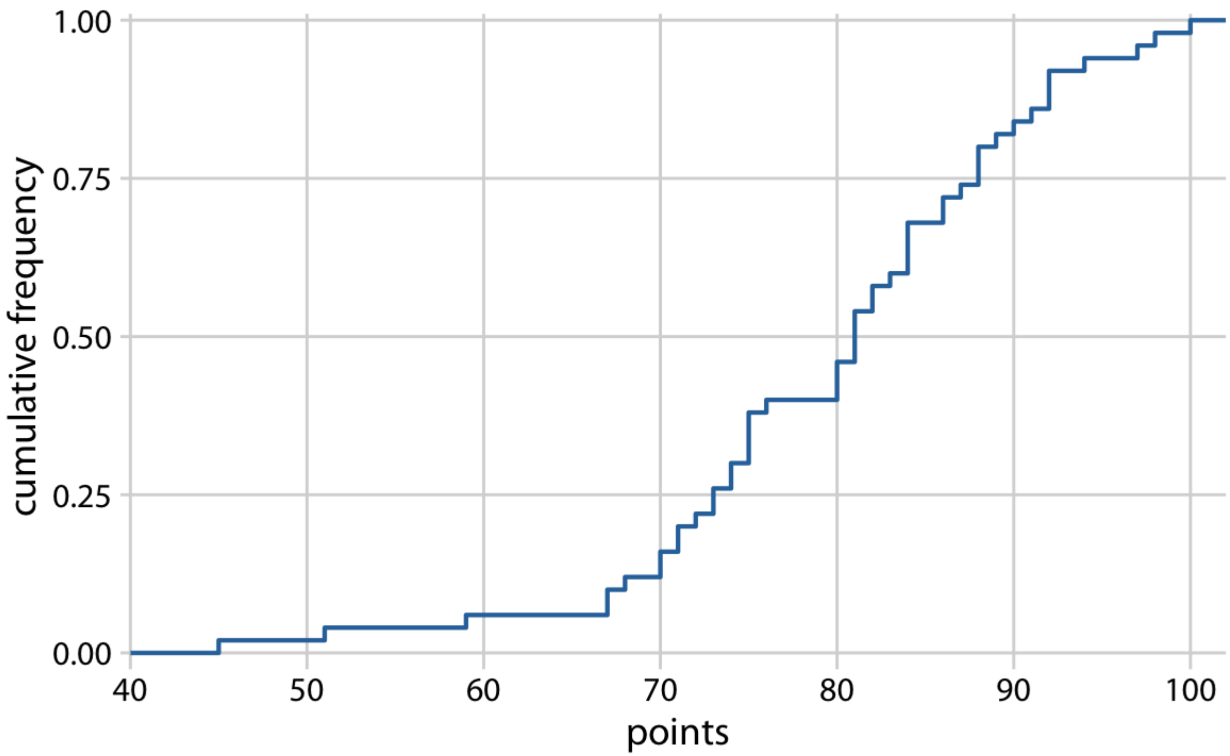
Visualization of Distributions: ECDFs

- Both are important, but ascending are used by default



Visualization of Distributions: ECDFs

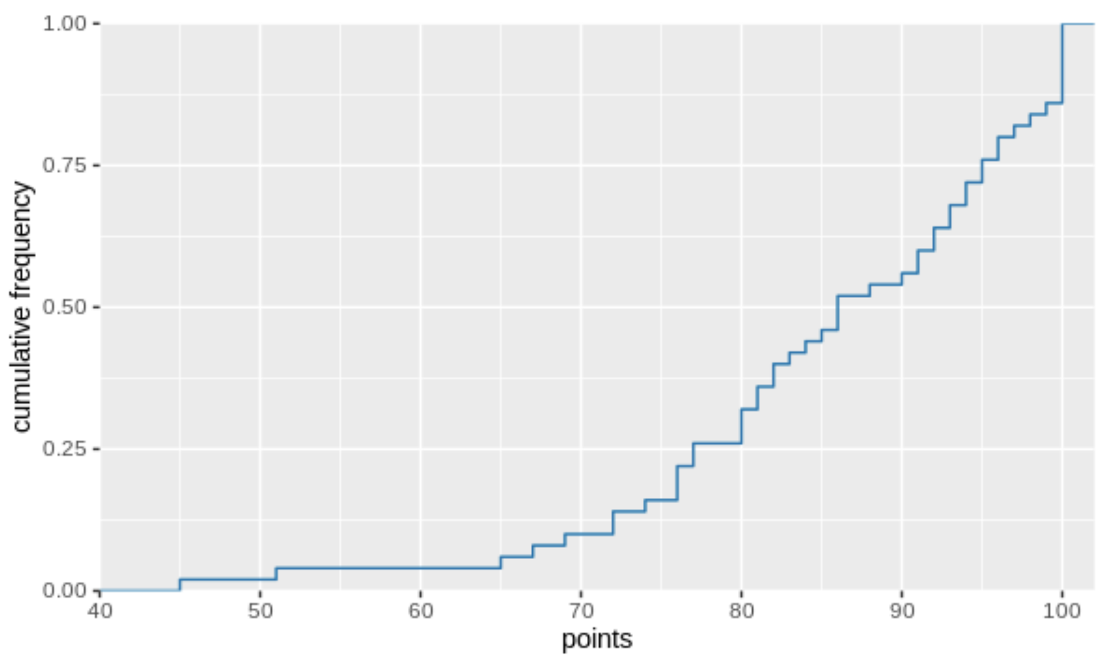
- Most commonly adopted choices for the visualization:
  - Individual points are not shown
  - Y-axis represent the cumulative frequency



Visualization of Distributions: ECDFs

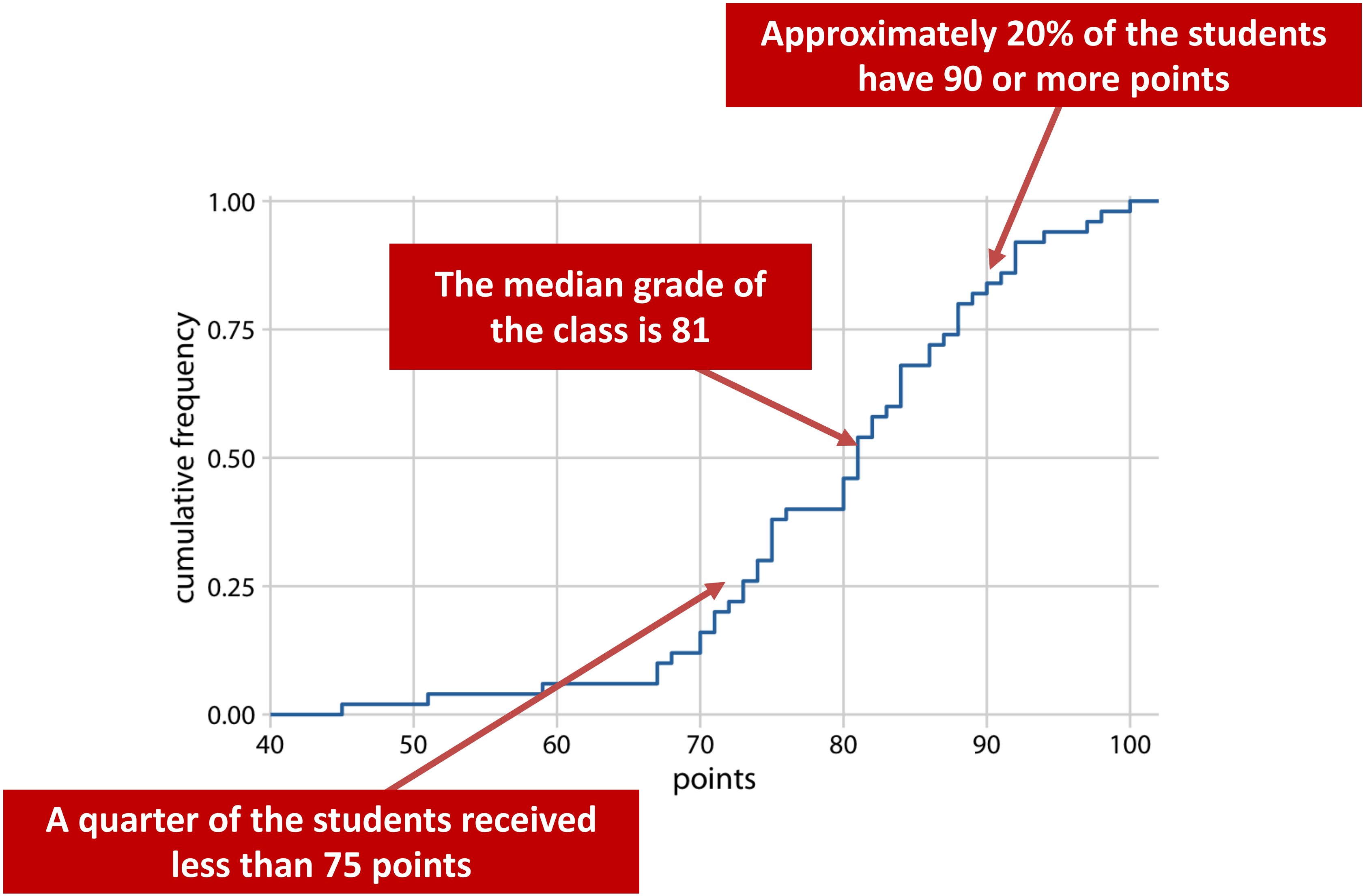
- How to create in ggplot2?
  - Just remove the call to geom_point() and do not multiply y by the total

```
ggplot(student_data, aes(x = points, y = ..y..)) +
    stat_ecdf(geom = "step", color = "#0072B2") +
    scale_x_continuous(limits = c(40, 102), expand = c(0, 0),
                       breaks = 10*(4:10)) +
    scale_y_continuous(limits = c(0, 1), expand = c(0, 0),
                       name = "cumulative frequency")
```



Visualization of Distributions: ECDFs

- After you learn it exists, it is simple to interpret!



Visualization of Distributions: ECDFs

- The case of skewed distributions
  - **Existence of heavy tails make visualization challenging**
  - E.g. number of people living in diferente cities, contacts in social networks, word frequencies (Zipf’s law!), richness in the world, etc.
  - Decay is slower than exponential
    - Mean of the distribution is small, but there are many large values
  - *Example* of skewed distribution: *power-law*

Visualization of Distributions: ECDFs

- Example: dataset of populations of US counties

```
# read data
counties <- read.csv("https://www.ics.uci.edu/~algol/
teaching/informatics143w2021/counties.csv")
```

pp | counties

Filter

	X	county_name	state	state_fips	county_fips	fips_class	CSA	CBSA	population
1	1	Autauga County	AL	1	1	H1	NA	33860	54571
2	2	Baldwin County	AL	1	3	H1	380	19300	182265
3	3	Barbour County	AL	1	5	H1	NA	NA	27457
4	4	Bibb County	AL	1	7	H1	142	13820	22915
5	5	Blount County	AL	1	9	H1	142	13820	57322
6	6	Bullock County	AL	1	11	H1	NA	NA	10914
7	7	Butler County	AL	1	13	H1	NA	NA	20947
8	8	Calhoun County	AL	1	15	H1	NA	11500	118572
9	9	Chambers County	AL	1	17	H1	194	46740	34215
10	10	Cherokee County	AL	1	19	H1	NA	NA	25989

Showing 1 to 11 of 3,235 entries, 9 total columns

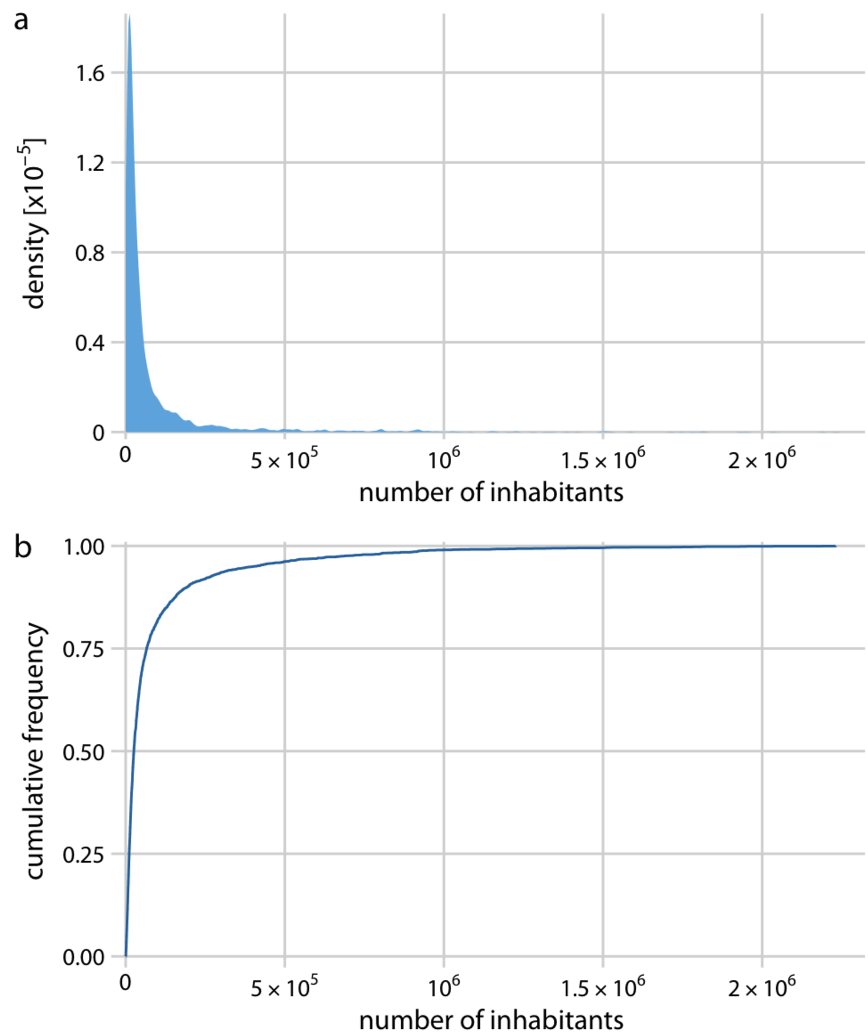
Console | Terminal | Jobs

/home/ics-home/

```
> counties <- read.csv("~/public_html/teaching/informatics143w2021/counties.csv")
> View(counties)
```

Visualization of Distributions: ECDFs

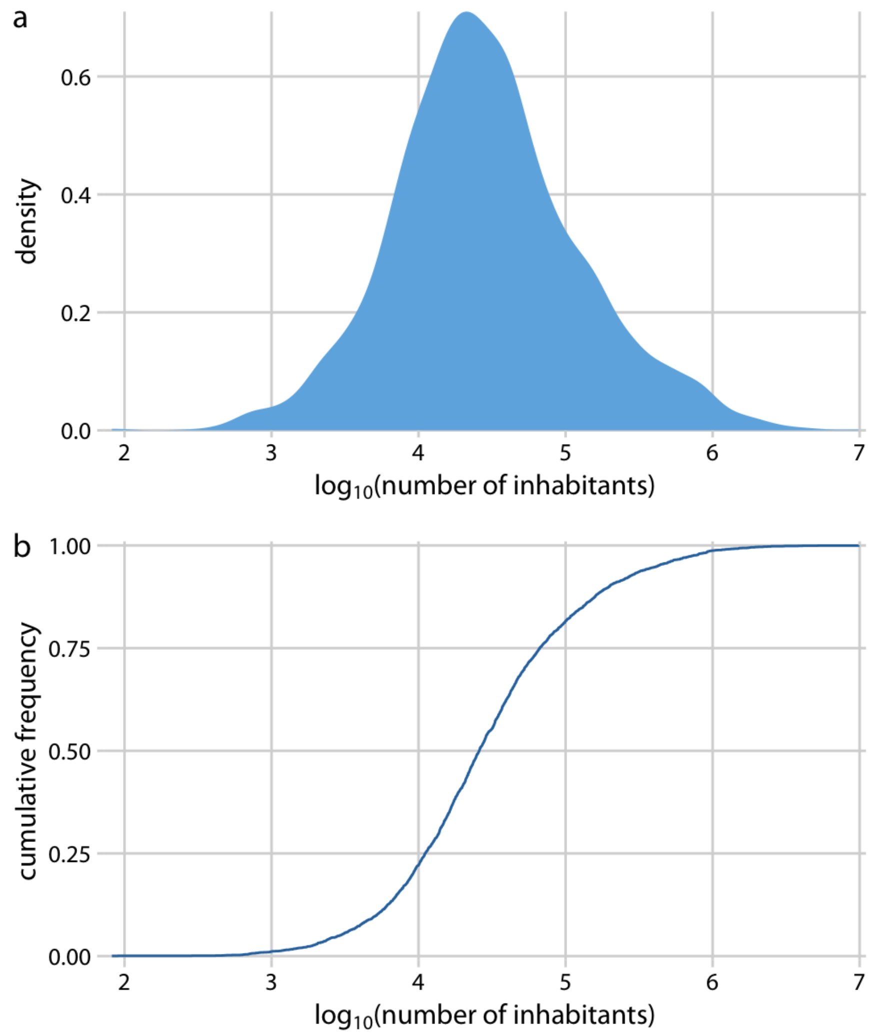
- Example: dataset of populations of US counties



Hard to see useful information

Visualization of Distributions: ECDFs

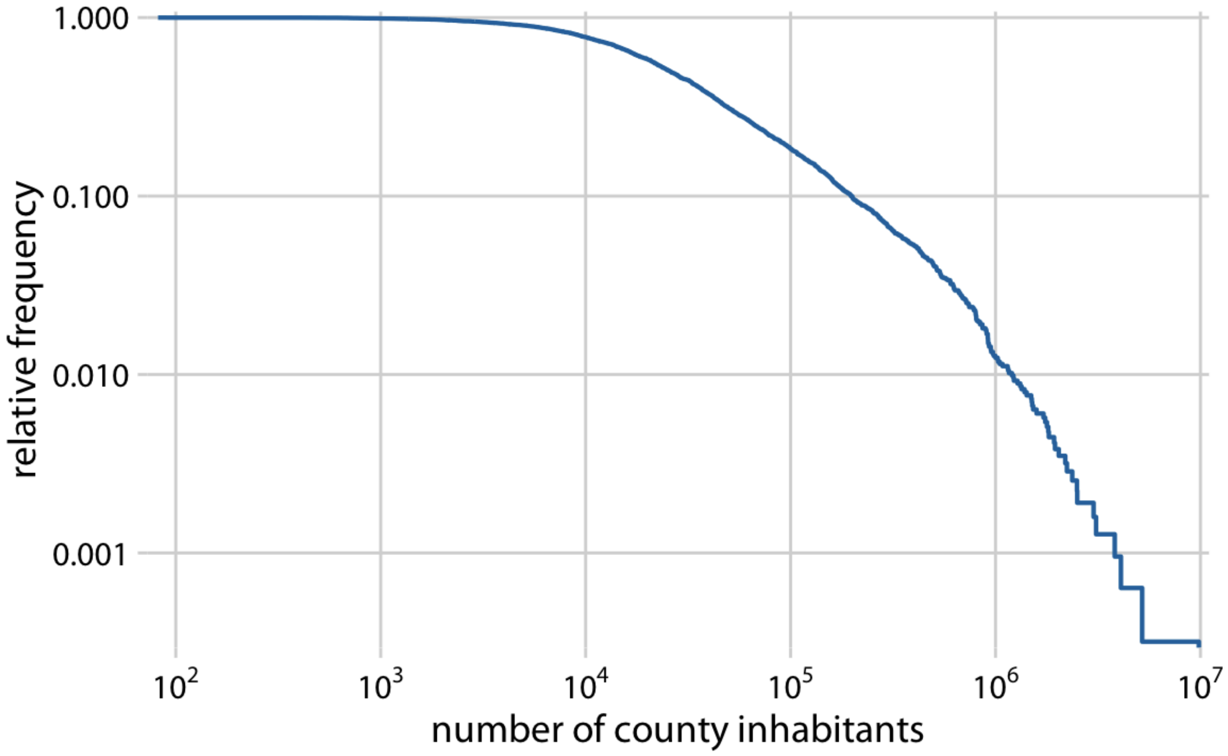
- Example: dataset of populations of US counties



Much better

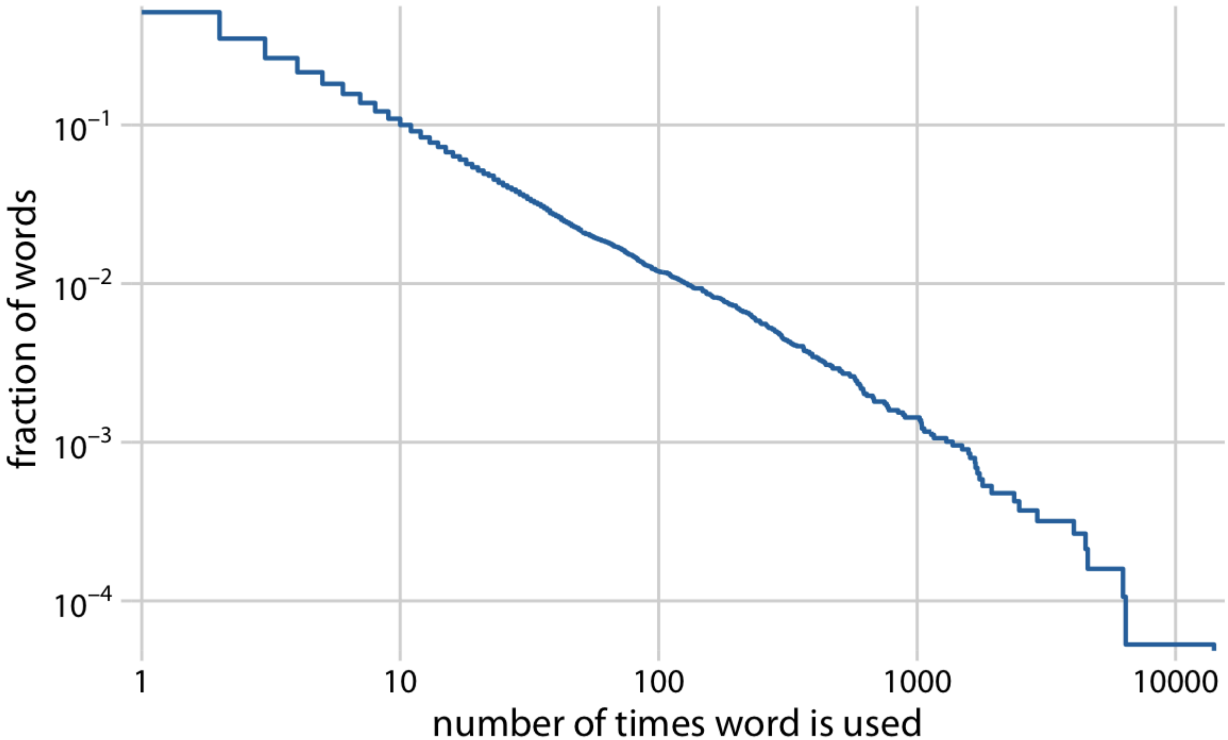
Visualization of Distributions: ECDFs

- Example: dataset of populations of US counties in log-log



Visualization of Distributions: ECDFs

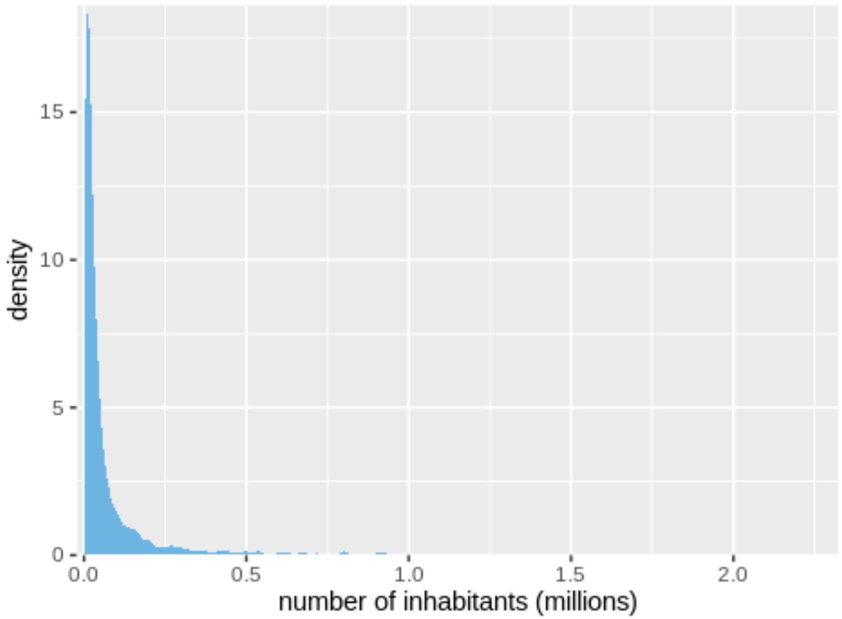
- Example: word frequency distribution in Moby Dick



Visualization of Distributions: ECDFs

- Bad example: dataset of populations of US counties

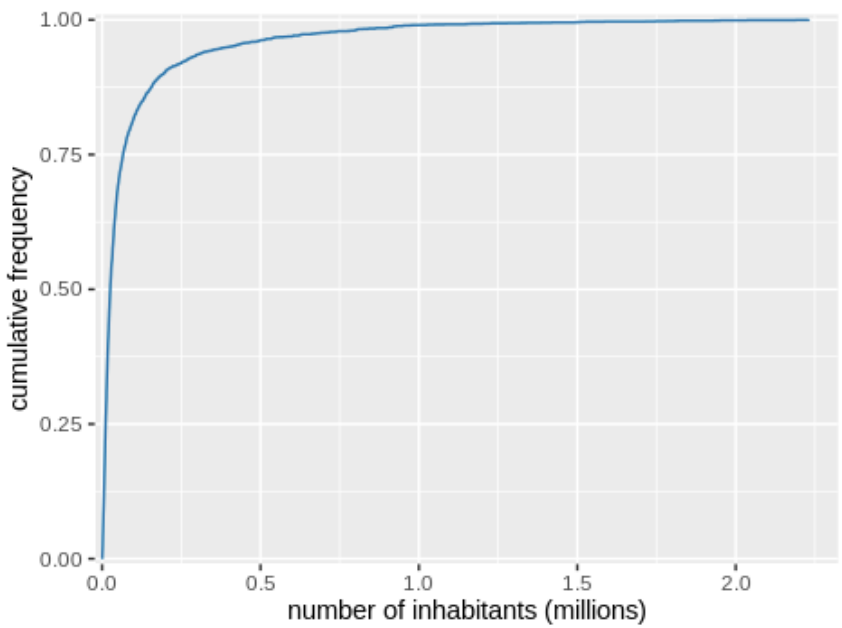
```
ggplot(counties, aes(x=population/1e6)) +
    geom_density(fill = "#56B4E9", color = "transparent") +
    scale_x_continuous(expand = c(0.01, 0),
                       name = "number of inhabitants (millions)",
                       limits = c(0, 2.3),
                       breaks = 0.5*(0:4)) +
    scale_y_continuous(expand = c(0, 0))
```



Visualization of Distributions: ECDFs

- Bad example: dataset of populations of US counties

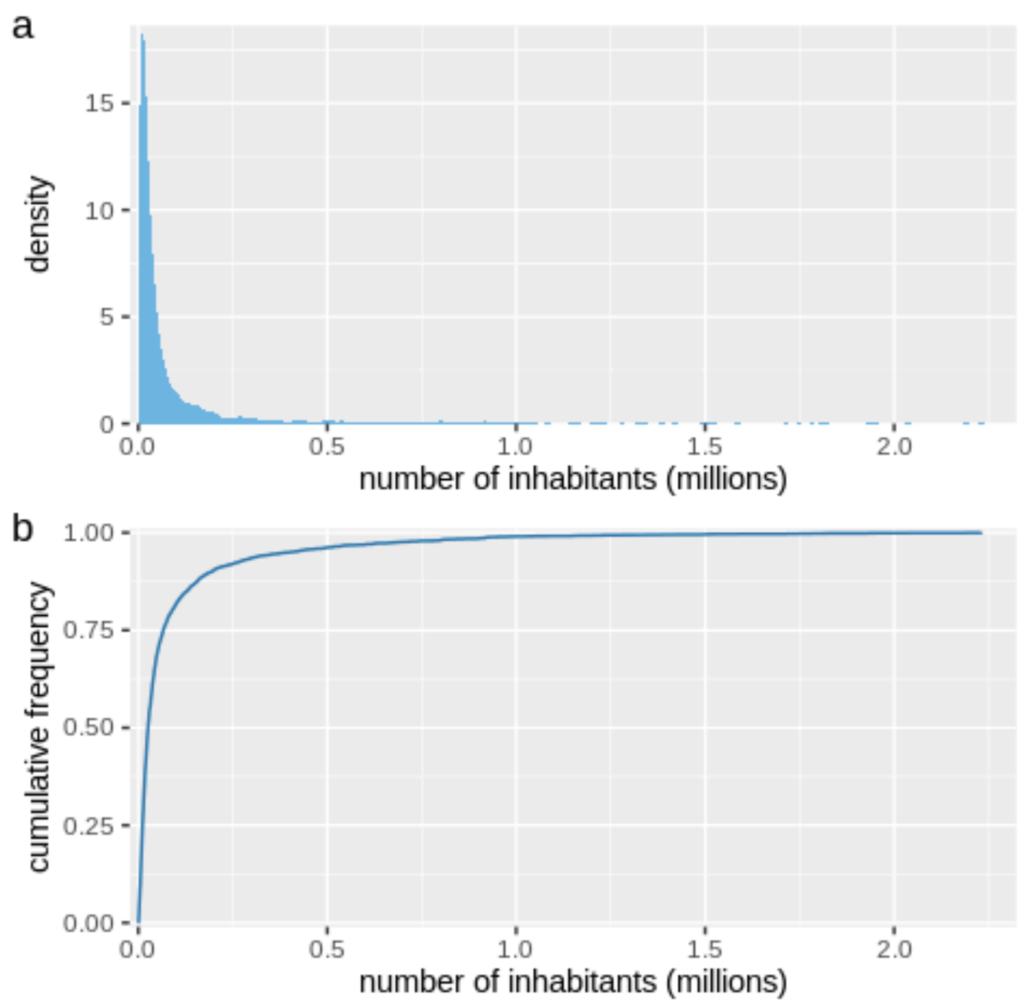
```
ggplot(counties, aes(x=population/1e6)) +
      stat_ecdf(geom = "step", color = "#0072B2", pad = FALSE) +
      scale_x_continuous(expand = c(0.01, 0),
                         name = "number of inhabitants (millions)",
                         limits = c(0, 2.3),
                         breaks = 0.5*(0:4)) +
      scale_y_continuous(expand = c(0.01, 0),
                         name = "cumulative frequency")
```



Visualization of Distributions: ECDFs

- Bad example: cowplot::plot_grid() combining plots

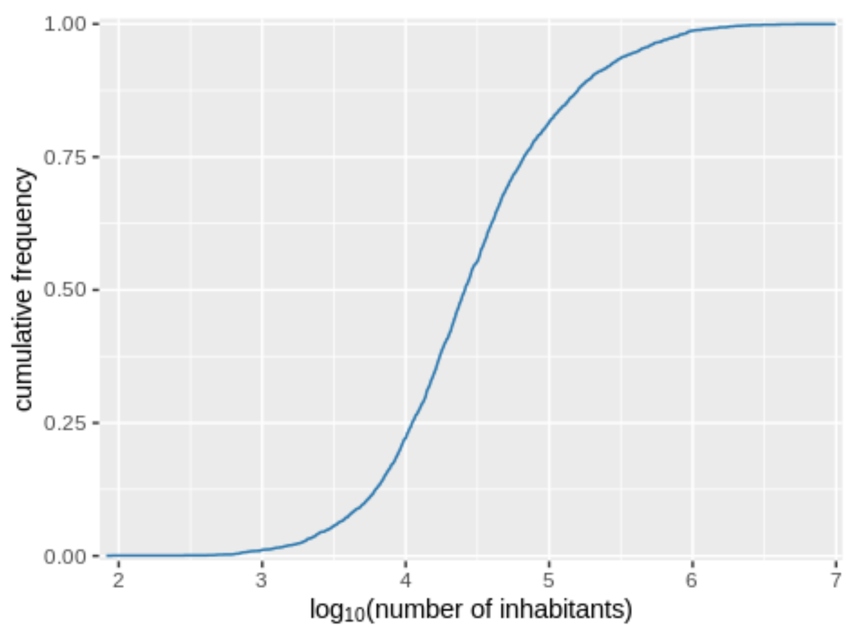
```
p1 <- ggplot(counties, aes(x=population/1e6)) +
      geom_density(fill = "#56B4E9",
         color = "transparent") +
      scale_x_continuous(expand = c(0.01, 0),
         name = "number of inhabitants (millions)",
         limits = c(0, 2.3),
         breaks = 0.5*(0:4)) +
      scale_y_continuous(expand = c(0, 0))

p2 <- ggplot(counties, aes(x=population/1e6)) +
       stat_ecdf(geom = "step", color = "#0072B2",
         pad = FALSE) +
       scale_x_continuous(expand = c(0.01, 0),
         name = "number of inhabitants (millions)",
         limits = c(0, 2.3),
         breaks = 0.5*(0:4)) +
       scale_y_continuous(expand = c(0.01, 0),
         name = "cumulative frequency")

cowplot::plot_grid(p1, p2,
         ncol = 1, align = 'v', labels = 'auto',
         label_fontface = "plain", hjust = 0,
         vjust = 1)
```



Visualization of Distributions: ECDFs

- Example: dataset of populations of US counties, log transformed

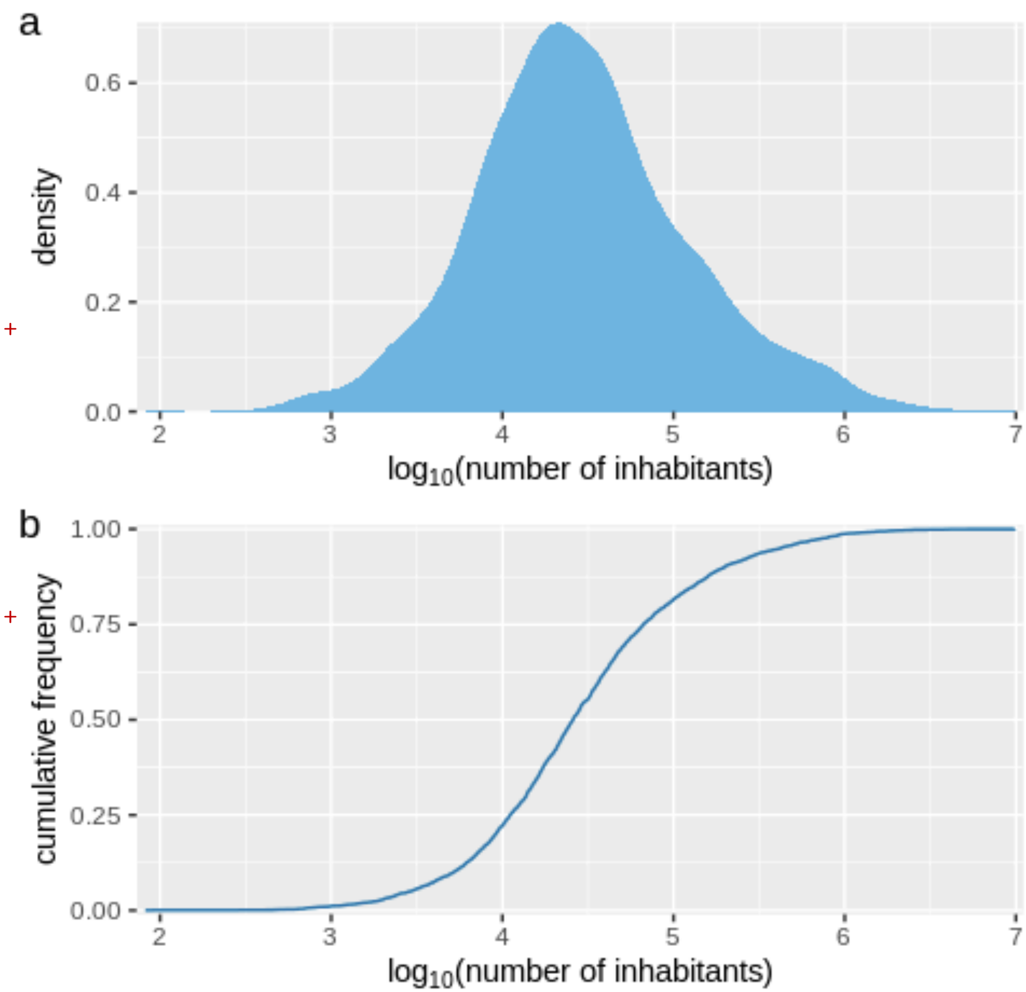
```
ggplot(counties, aes(x=log10(population))) +
  stat_ecdf(geom = "step", color = "#0072B2", pad = FALSE) +
  scale_x_continuous(
    expand = c(0.01, 0),
    name = expression(paste("log"["10"], "(number of inhabitants)"))) +
  scale_y_continuous(expand = c(0.01, 0), name = "cumulative frequency")
```



Visualization of Distributions: ECDFs

- Example: dataset of populations of US counties, log transformed

```
p1 <- ggplot(counties, aes(x=log10(population))) +
        geom_density(fill = "#56B4E9",
                     color = "transparent") +
        scale_x_continuous(
              expand = c(0.01, 0),
              name = expression(paste("log"["10"],
                     "(number of inhabitants)"))) +
        scale_y_continuous(expand = c(0, 0),
              name = "density")

p2 <- ggplot(counties, aes(x=log10(population))) +
        stat_ecdf(geom = "step",
              color = "#0072B2", pad = FALSE) +
        scale_x_continuous(
              expand = c(0.01, 0),
              name = expression(paste("log"["10"],
                     "(number of inhabitants)"))) +
        scale_y_continuous(expand = c(0.01, 0),
              name = "cumulative frequency")

cowplot::plot_grid(p1, p2, ncol = 1,
        align = 'v', labels = 'auto’,
        label_fontface = "plain",
        hjust = 0, vjust = 1)
```



Visualization of Distributions: ECDFs

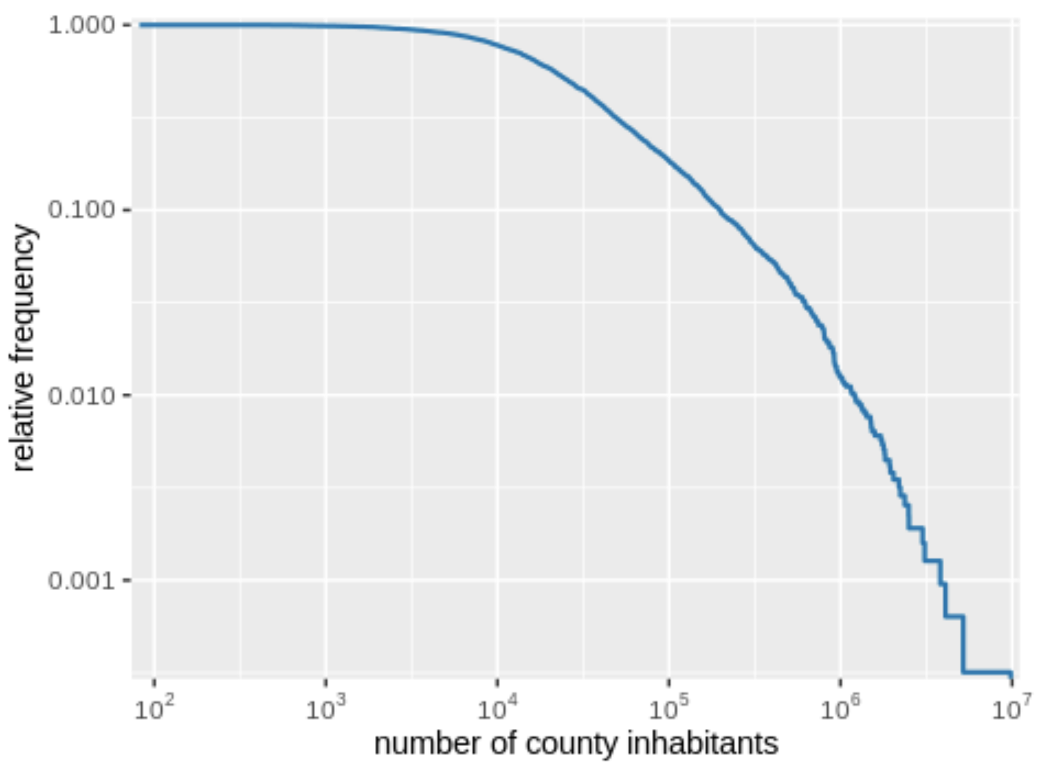
- Example: ECDF of populations of US counties in log-log

```
ggplot(counties, aes(x=population, y = 1-..y..)) +
  stat_ecdf(geom = "step", color = "#0072B2", size = 0.75, pad = FALSE) +
  scale_x_log10(expand = c(0.01, 0),
                breaks = c(1e2, 1e3, 1e4, 1e5, 1e6, 1e7),
                labels = c(expression(10^2),
                           expression(10^3), expression(10^4),
                           (10^5), expression(10^6),
                           expressioexpressionn(10^7)),
                name = "number of county inhabitants") +
  scale_y_log10(expand = c(0.01, 0),
                breaks = c(1e-3, 1e-2, 1e-1, 1),
                name = "relative frequency")
```

Visualization of Distributions: ECDFs

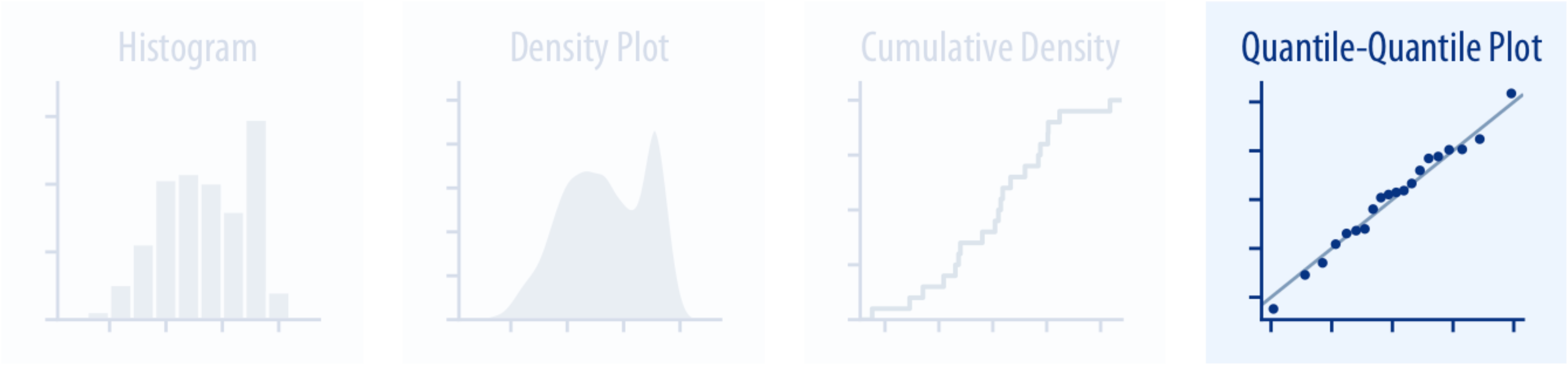
- Example: ECDF of populations of US counties in log-log

```
ggplot(counties, aes(x=population, y = 1-..y..)) +
  stat_ecdf(geom = "step", color = "#0072B2", size = 0.75, pad = FALSE) +
  scale_x_log10( <see previous slide> ) +
  scale_y_log10( <see previous slide> )
```



Visualization of Distributions

- Quantile-quantile plots (or Q-Q plots)

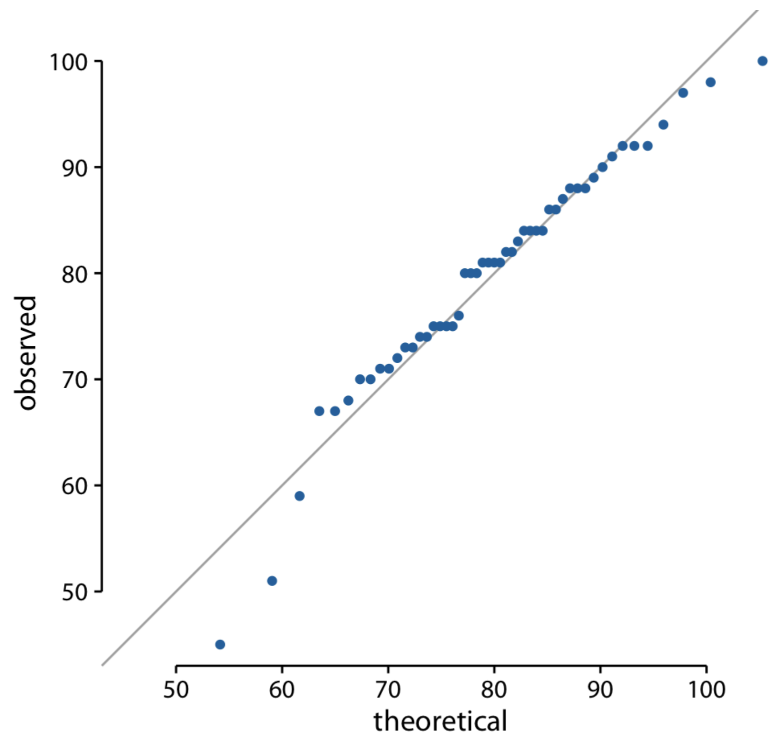


Visualization of Distributions: Q-Q plots

- Quantile-quantile plots
  - **Used to determine if data follow or not a certain distribution**
  - Just as ECDFs they are based on ranking data points and visualizing the relation between data and rank
  - Ranks are used to *predict* where a point should fall *if the data follows* the reference *distribution*
  - The Normal distribution is widely used as reference

Visualization of Distributions: Q-Q plots

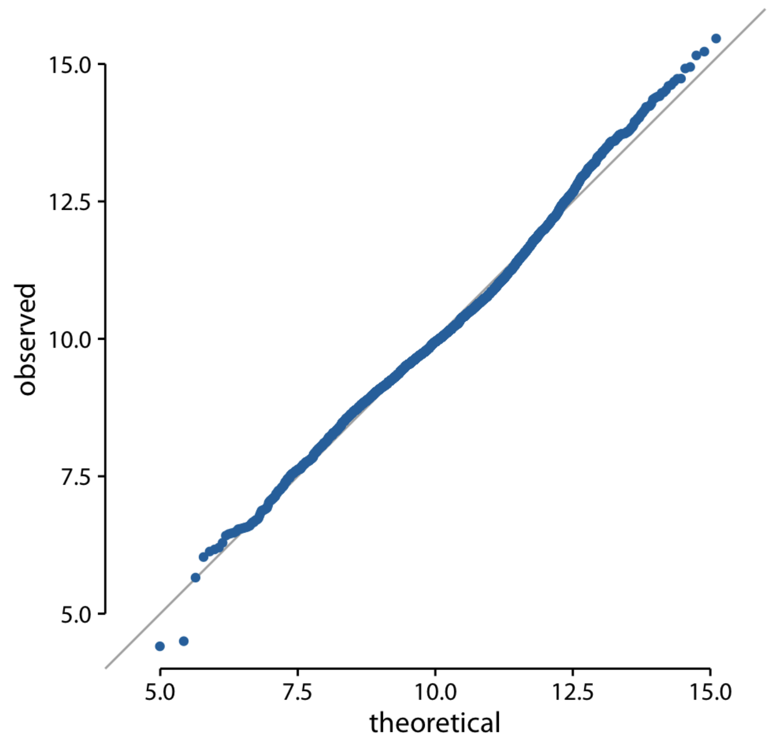
- Example: Q-Q plot of the dataset of student grades



- Reference: normal distribution

Visualization of Distributions: Q-Q plots

- Example: Q-Q plot of the number of inhabitants in US counties



- Reference: log-normal distribution

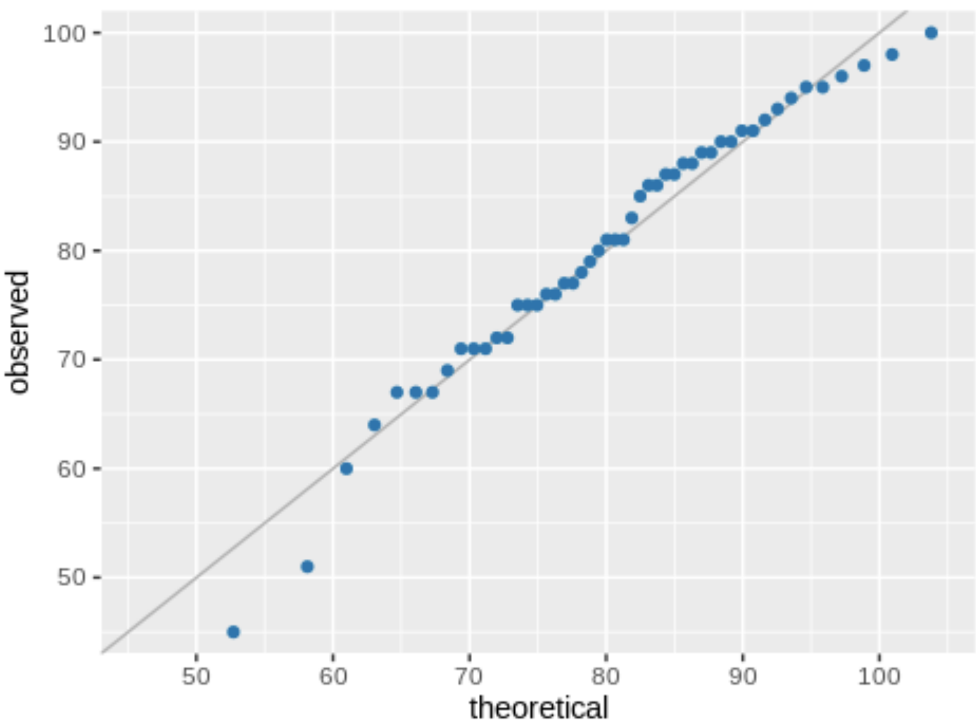
Visualization of Distributions: Q-Q plots

- How to create in ggplot2?
  - Use the stat_qq()
  - Q-Q plot of the dataset of student grade using the normal as reference.

```
# Theorethical distribution parameter estimate
params <- as.list(MASS::fitdistr(student_data$points, "normal")$estimate)

# QQ plot
ggplot(student_data, aes(sample = points)) +
  geom_abline(slope = 1, intercept = 0, color = "grey70") +
  stat_qq(dparams = params, color = "#0072B2") +
  scale_x_continuous(limits = c(43, 107), expand = c(0, 0),
                     breaks = 10*(5:10)) +
  scale_y_continuous(limits = c(43, 102), expand = c(0, 0),
                     breaks = 10*(5:10), name = "observed")
```

Visualization of Distributions: Q-Q plots

- How to create in ggplot2?
  - Use the stat_qq()
  - Q-Q plot of the dataset of student grade using the normal as reference.



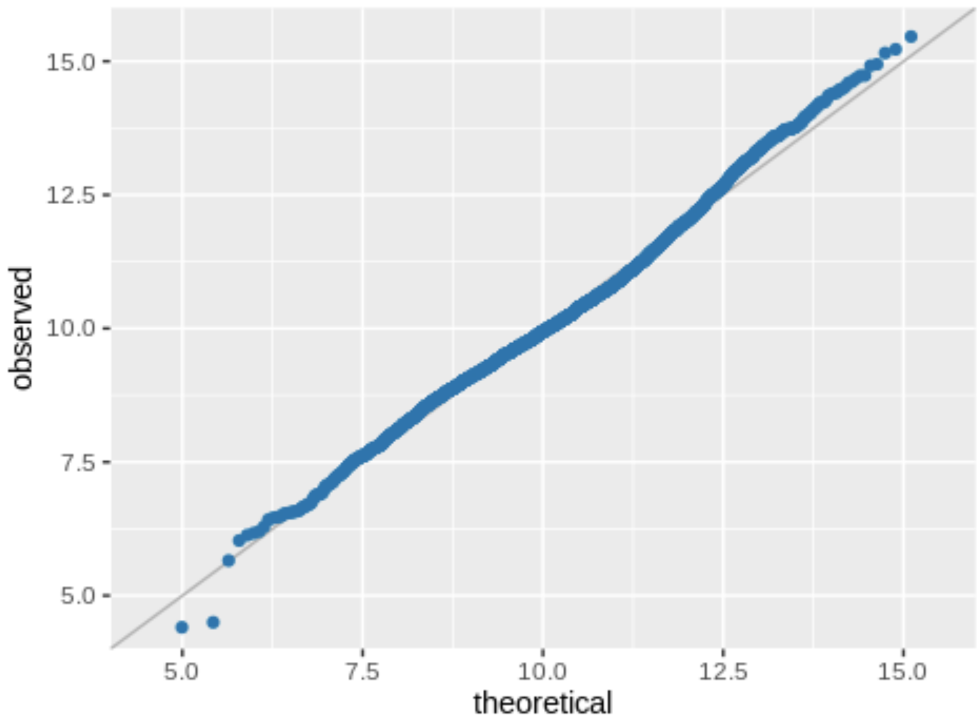
Visualization of Distributions: Q-Q plots

- How to use a log-normal as reference?
  - Just take the log of the data
  - Q-Q plot of the dataset of populations of the US counties using the log-normal as reference.

```
params <- as.list(
     MASS::fitdistr(na.omit(log(counties$population)), "normal")$estimate)

ggplot(counties, aes(sample = log(population))) +
  geom_abline(slope = 1, intercept = 0, color = "grey70") +
  stat_qq(dparams = params, color = "#0072B2") +
  scale_x_continuous(limits = c(4, 16), expand = c(0, 0),
                     breaks = 5+2.5*(0:4)) +
  scale_y_continuous(limits = c(4, 16), expand = c(0, 0),
                     name = "observed", breaks = 5+2.5*(0:4))
```

Visualization of Distributions: Q-Q plots

- How to use a log-normal as reference?
  - Just take the log of the data
  - Q-Q plot of the dataset of populations of the US counties using the log-normal as reference.



Visualization of Distributions: multiple distributions

- What if you need to visualize more than one distribution at once, where **now** *more* is a large number?

- What to do in these cases?
  – E.g. how temperature varies along different months *and* within each month?

Visualization of Distributions: multiple distributions

- Some options...

Visualization of Distributions: multiple distributions

- What if you need to visualize more than one distribution at once, where **now** *more* is a large number?

- What to do in these cases?
  - E.g. how temperature varies along different months *and* within each month?

- Helpful to think in terms of:
  - *Response variable*: the variable whose distributions you want to study
  - One or more *grouping variables*: variables that define data subsets

Visualization of Distributions: multiple distributions

- What if you need to visualize more than one distribution at once, where **now** *more* is a large number?

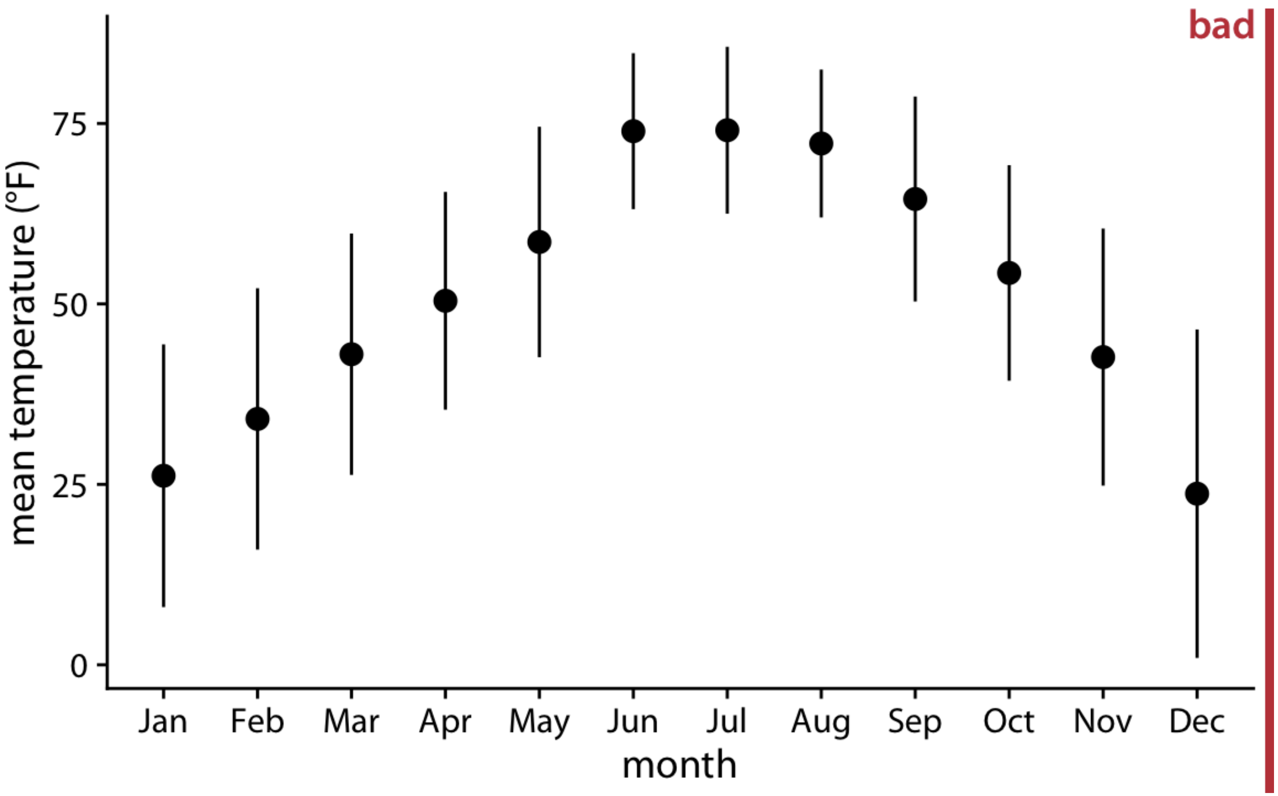
- What to do in these cases?
  - E.g. how temperature varies along different months *and* within each month?

- Helpful to think in terms of:
  - *Response variable*: the variable whose distributions you want to study
  - One or more *grouping variables*: variables that define data subsets

- Response variable will be mapped along one axis, the grouping variable along the other

Visualization of Distributions: Simplest approach

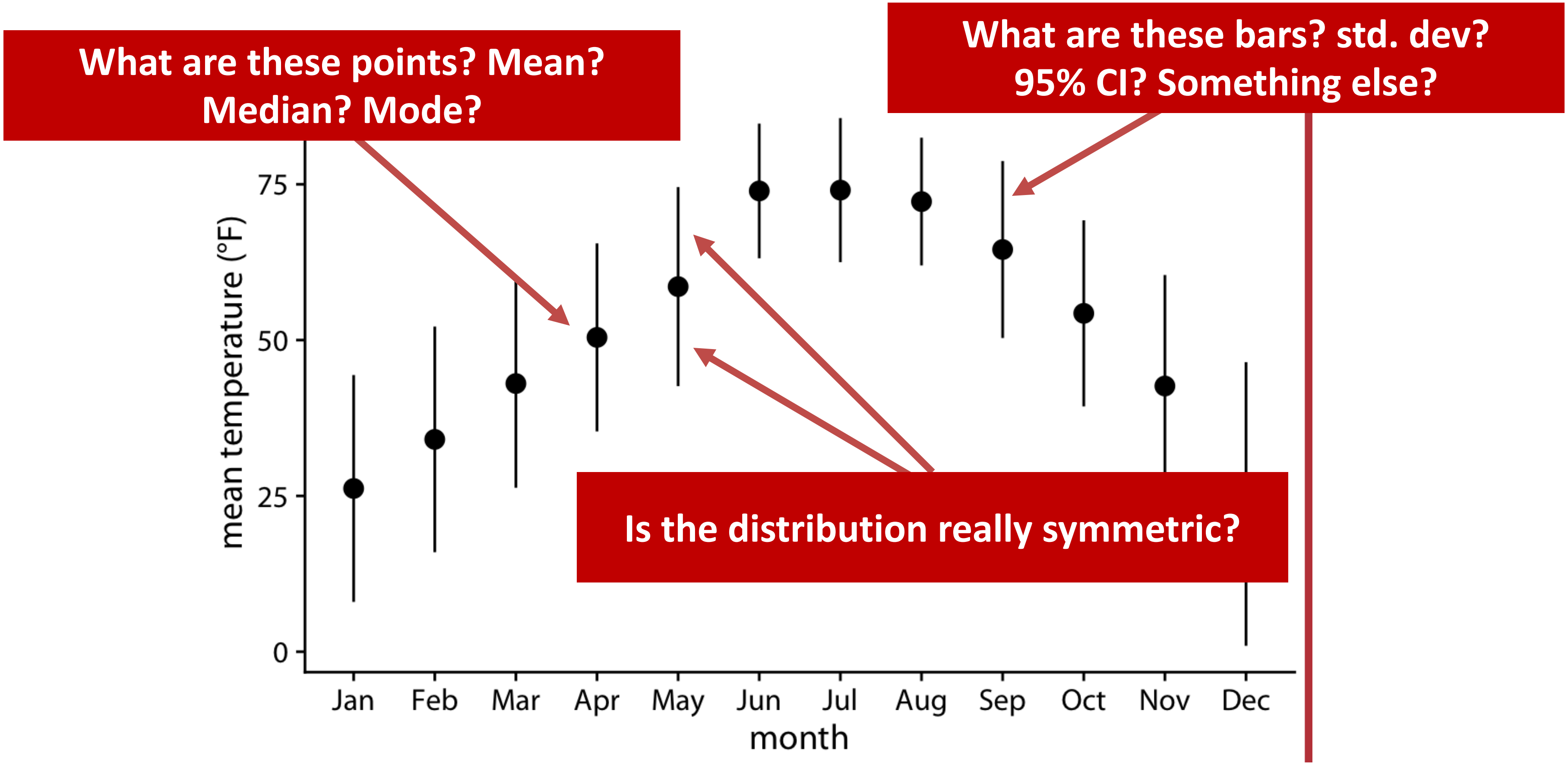
- Divide the data into subsets using the grouping variable
- Compute mean/medians/mode and some dispersion metric for each subset
- Display points at the mean/median/mode and add the dispersion metric as a bar



Data source: Weather Underground

Visualization of Distributions: Simplest approach

- Divide the data into subsets using the grouping variable
- Compute mean/medians/mode and some dispersion metric for each subset
- Display points at the mean/median/mode and add the dispersion metric as a bar



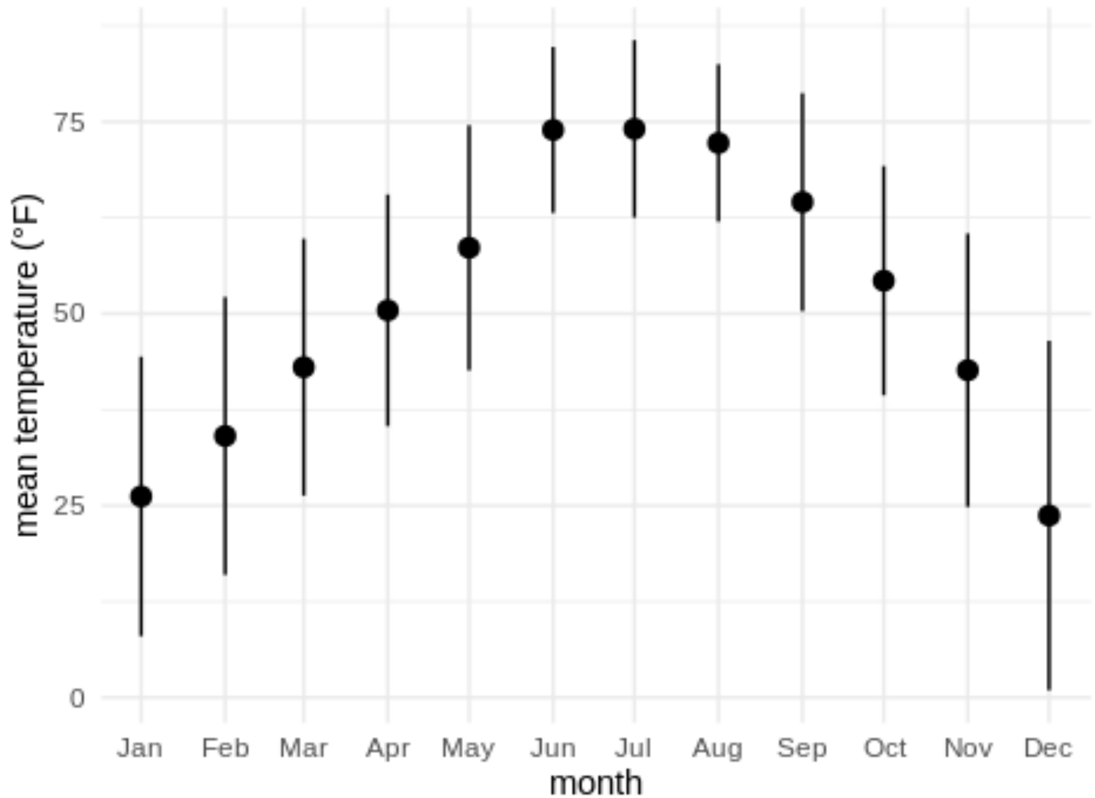
Visualization of Distributions: Simplest approach

- How to do it in ggplot2?
  - Use stat_summary()

```
lincoln_df <-read.csv(
"https://www.ics.uci.edu/~algol/teaching/informatics143w2021/lincoln_df.csv")

ggplot(lincoln_df, aes(x = month_short, y = Mean.Temperature..F.)) +
  stat_summary(
    fun = mean,
    fun.max = function(x) {mean(x) + 2*sd(x)},
    fun.min = function(x) {mean(x) - 2*sd(x)},
    geom = "pointrange"
  ) +
  xlab("month") +
  ylab("mean temperature (°F)") +
  scale_x_discrete(limits = lincoln_df$month_short,
                   expand = c(0.04,0, -0.04, 0)) +
  theme_minimal()
```

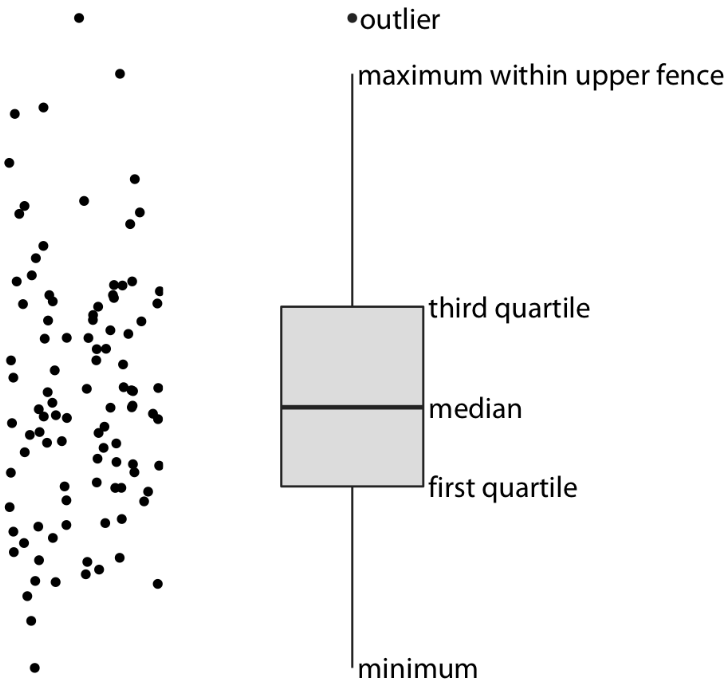
Visualization of Distributions: Simplest approach

- How to do it in ggplot2?
  - Use stat_summary()



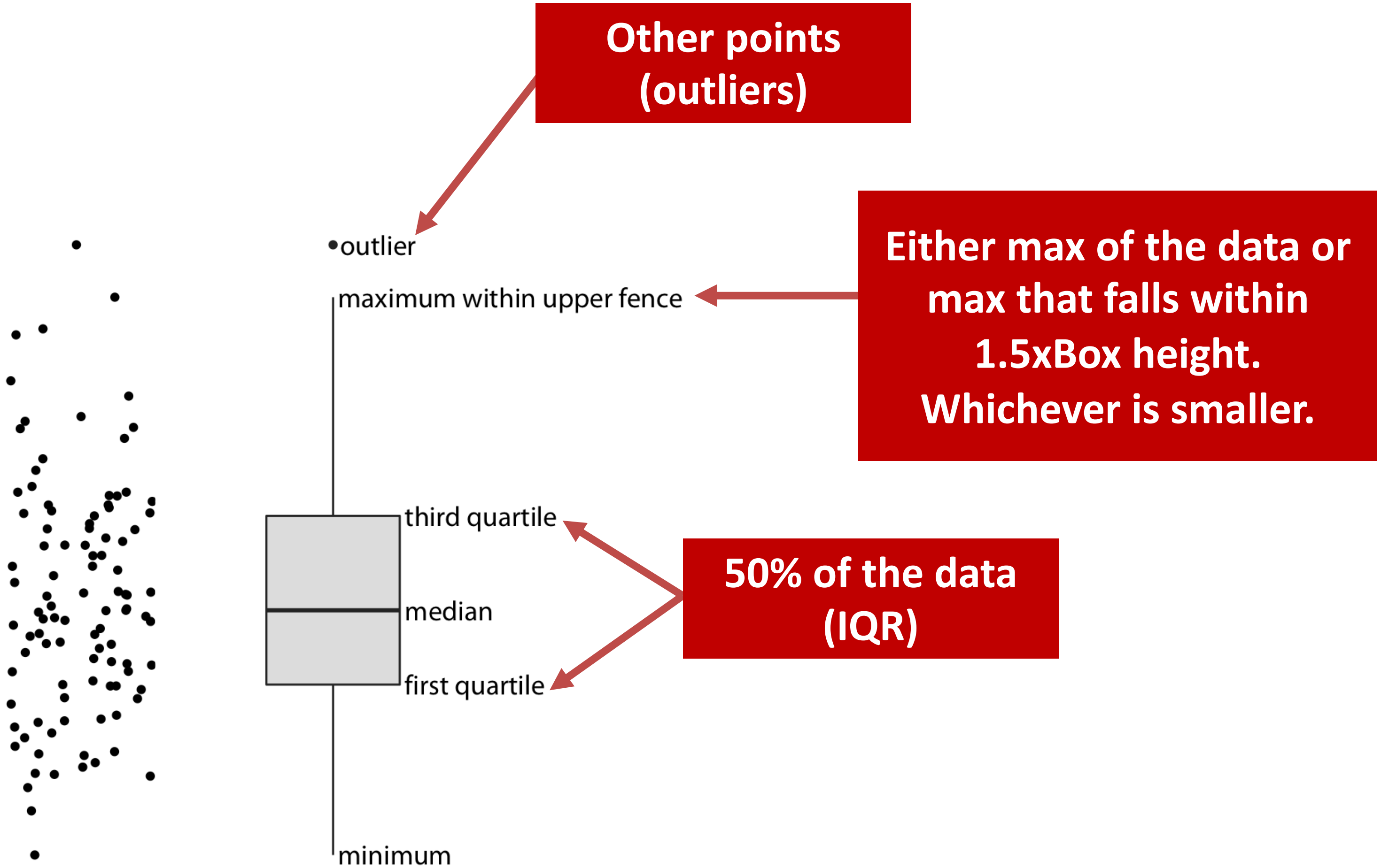
Visualization of Distributions: Boxplot

- Boxplots
  - Divide the data into quartiles and represent



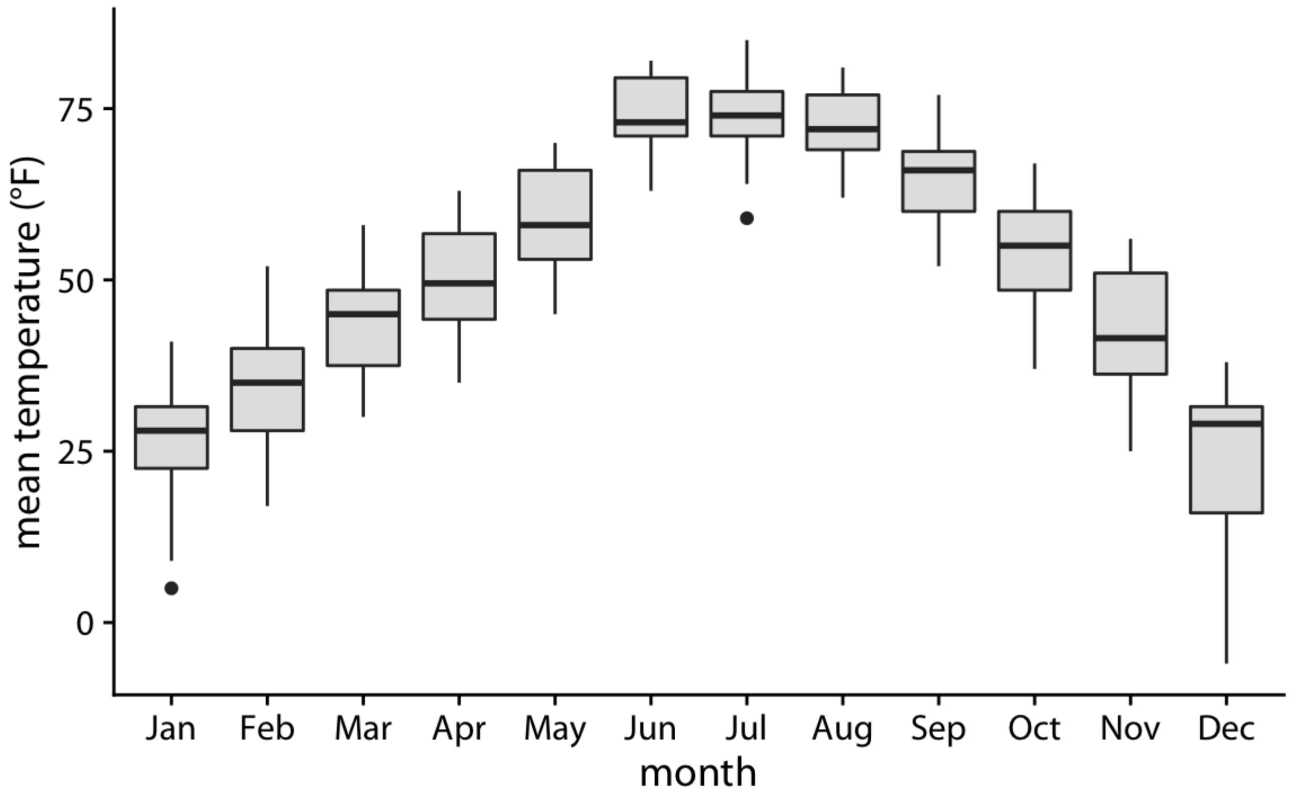
Visualization of Distributions: Boxplot

- Boxplots
  - Divide the data into quartiles and represent



Visualization of Distributions: Boxplot

- Boxplot of the temperature dataset



Data source: Weather Underground

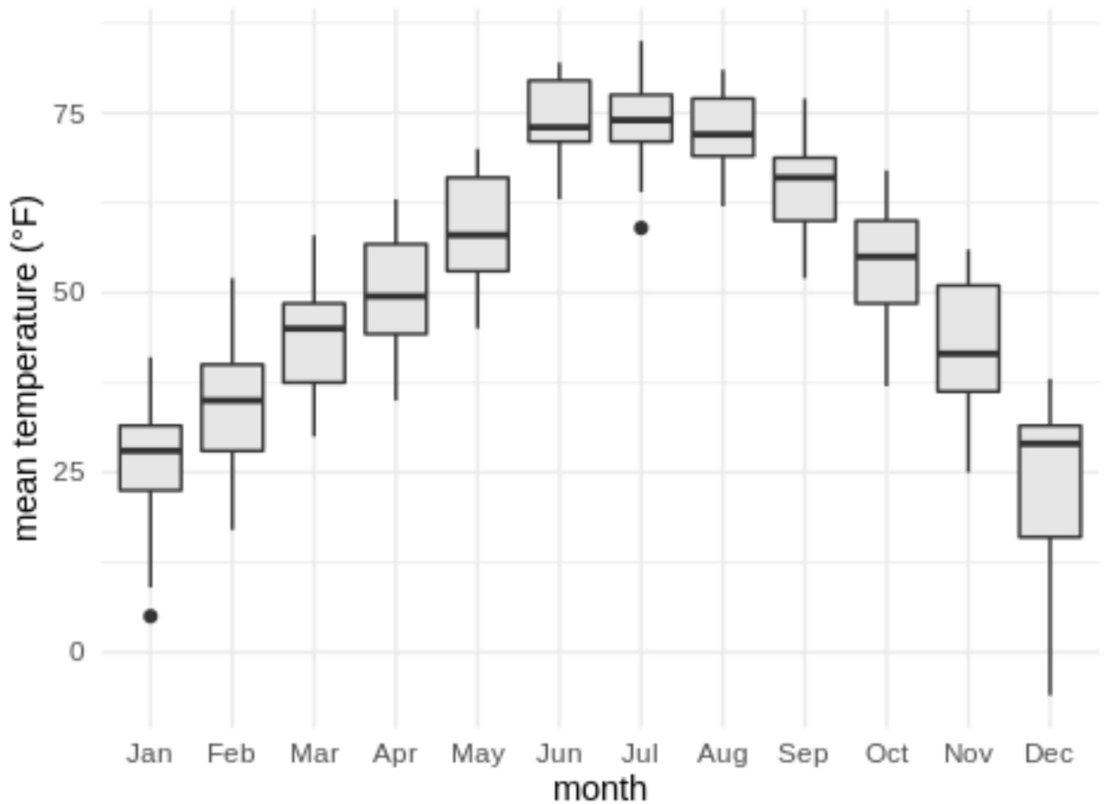
Visualization of Distributions: Boxplot

- How to do it in ggplot2?
  - Use geom_boxplot()

```
lincoln_df <-read.csv(
"https://www.ics.uci.edu/~algol/teaching/informatics143w2021/lincoln_df.csv")

lincoln_df$month_short <- factor(lincoln_df$month_short,
                                 levels = unique(lincoln_df$month_short))

ggplot(lincoln_df, aes(x = month_short, y = Mean.Temperature..F.)) +
  geom_boxplot(fill = 'grey90') +
  xlab("month") +
  ylab("mean temperature (°F)") +
  theme_minimal()
```

Visualization of Distributions: Boxplot

- How to do it in ggplot2?
  - Use geom_boxplot()



Visualization of Distributions: Boxplot

- Boxplots
  - Invented betweeen 50-70’s by Mary Eleanor Spear and John Tukey
  - Very simple to read
  - Very simple to draw **by hand**
    - Very important aspect until recent times...
  - But they are still *hiding* information.