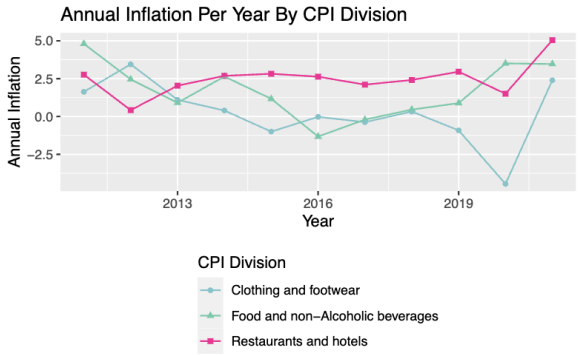
```r
{r}
#| label: annual-inflation-vs-year

my_palette2 <- c('cadetblue3', 'aquamarine3', 'deeppink')


ggplot(data = DOI_joined_USInflation_CPI, aes(x = year, y = annual_inflation,
       shape = description, color = description)) +
  geom_point() +
  geom_line() +
  labs(
    title = "Annual Inflation Per Year By CPI Division",
    x = "Year",
    y = "Annual Inflation",
    color = "CPI Division",
    shape = "CPI Division"
    ) +
  scale_color_manual(values = my_palette2) +
  theme(
        legend.position = "bottom",
        legend.direction = "vertical"
        )
```



Pivot

```r
{r}
#| label: country-inflation-longer

pivoted_CI <- country_inflation |>
  pivot_longer(
    cols = !country, names_to = "year", values_to = "annual_inflation",
    names_transform = as.numeric)
```

	country	1993	1994	1995
Austria				
	ustralia	1.753653	1.9696348	4.6277666
2	Austria	3.631786	2.9534065	2.2433638
3	Belgium	2.754426	2.3775445	1.4679612
4	Canada	1.865079	0.1655629	2.1487603

-->

country <chr>	year <dbl>	annual_inflation <dbl>
Australia	1993	1.753653e+00
Australia	1994	1.969635e+00

```r
patients
```

```
# A tibble: 3 × 4
  patient_id pulse_1 pulse_2 pulse_3
  <chr>        <dbl>   <dbl>   <dbl>
1 XYZ             70      85      73
2 ABC             90      95     102
3 DEF            100      80      70
```

```r
patients_longer <- patients |>
  pivot_longer(
    cols = !patient_id,
    names_to = "measurement",
    values_to = "pulse_rate"
  )
```

Arrange/Select/Mutate

```r
country_inflation |>
  mutate(inf_ratio = `2021` / `1993`) |>
  arrange(desc(inf_ratio)) |>
  select(country, inf_ratio)
```

Filter

```r
pivoted_CI |>
  filter(annual_inflation == max(annual_inflation, na.rm = TRUE) |
         annual_inflation == min(annual_inflation, na.rm = TRUE))
```

Vector

```r
countries_of_interest <- c("Italy", "United States", "Japan")
```

Summarize / Quantile

```r
midwest |>
  summarize(
    median = median(popdensity),
    q1 = quantile(popdensity, 0.25),
    q3 = quantile(popdensity, 0.75)
  )
```

Group_by (Prop counties in urban areas in each state)

```r
midwest |>
  mutate(metro = if_else(inmetro == 1, "Yes", "No")) |>
  group_by(state, metro) |>
  summarise(count = n()) |>
  mutate(proportion_inmetro = count / sum(count)) |>
  filter(metro == "Yes")
```

```r
midwest <- midwest |>
  mutate(potential_outlier = if_else(percollege > 40 |
  percbelowpoverty == max(percbelowpoverty), "Yes", "No"))
```

Scatterplot

```r
ggplot(midwest, aes(x=percollege, y=percbelowpoverty, color = potential_outlier,
       shape=state)) +
  geom_point() +
  labs(
    x = "percentage of people with a college degree",
    y = "percentage of people below poverty",
    title = "% people with college degree vs people below poverty by state"
  )
```

Summarize

```r
state_poverty <- midwest |>
  group_by(state) |>
  summarize(mean_percbelowpoverty = mean(percbelowpoverty)) |>
  select(state, mean_percbelowpoverty)
```

```r
state_population <- midwest |>
  group_by(state) |>
  summarize(total_population = sum(poptotal))
```

```r
pop_summary <- population_continent |>
  group_by(continent) |>
  summarize(total_pop = sum(population)) |>
  arrange(desc(total_pop))
```

Count by two vars

```r
flights |>
  count(month, day) |>
  arrange(desc(n))
```

Left_Join (keep all left singles)

Right_Join (keep all right singles)

```r
patients_longer |>
  group_by(patient_id) |>
  summarize(mean_pulse = mean(pulse_rate))
```

ggplot(
  patients_longer,
  aes(x = measurement, y = pulse_rate, group = patient_id, color = patient_id)) +

Initializing

country_inflation <- read_csv("data/country-inflation.csv")
glimpse(country_inflation)
pull(country_inflation, var="country")

Arrange/Select/Slice

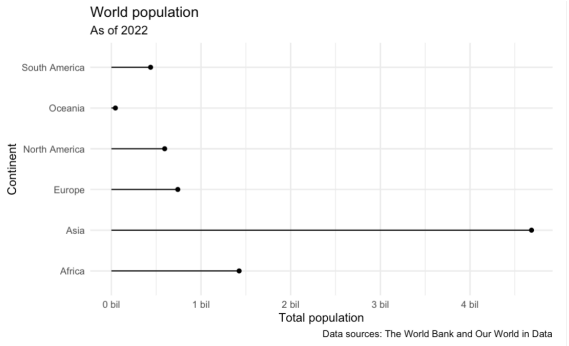
```r
country_inflation |>
  arrange(desc(`2021`)) |>
  select(country, `2021`) |>
  slice(0:3)
```

Logical operators

operator	definition
`<`	is less than?
`<=`	is less than or equal to?
`>`	is greater than?
`>=`	is greater than or equal to?
`==`	is exactly equal to?
`!=`	is not equal to?
`x & y`	is x AND y?
`x | y`	is x OR y?
`is.na(x)`	is x NA?
`!is.na(x)`	is x not NA?
`x %in% y`	is x in y?
`!(x %in% y)`	is x not in y?
`!x`	is not x? (only makes sense if `x` is `TRUE` or `FALSE`)

Visualization

```r
ggplot(population_summary) +
  geom_point(aes(x = total_pop, y = continent)) +
  geom_segment(aes(y = continent, yend = continent, x = 0, xend = total_pop)) +
  scale_x_continuous(labels = label_number(scale = 1/1000000, suffix = " bil")) +
  theme_minimal() +
  labs(
    x = "Total population",
    y = "Continent",
    title = "World population",
    subtitle = "As of 2022",
    caption = "Data sources: The World Bank and Our World in Data"
  )
```



Full_Join (keep all singles)
Inner_join (keep no singles, keep BOTH values)
Semi_join (keep no singles, keep left value)

```r
population_continent <- population |>
  left_join(continents, by = join_by(country == entity))
```

Case When

```r
{r}
#| label: data-clean

population_continent <- population |>
  mutate(
    country = case_when(
      country == "Congo, Dem. Rep." ~ "Democratic Republic of Congo",
      country == "Congo, Rep." ~ "Congo",
      country == "Hong Kong SAR, China" ~ "China",
      country == "Korea, Dem. People's Rep." ~ "North Korea",
      country == "Korea, Rep." ~ "South Korea",
      country == "Kyrgyz Republic" ~ "Kyrgyzstan",
      .default = country
      )
  ) |>
  left_join(continents, by = join_by(country == entity))

population_continent |> pull("continent") |> is.na() |> any()
#population_continent |> pull("continent") |> is.na()
```

Plots

- Pie charts and waffle charts are for visualizing distributions of categorical data only
- Scatterplots are for visualizing the relationship between two numerical variables
- Histograms, Density Plots, and Box Plots = visualizing the distributions of one variable

Types/Classes

double: a real number stored in double-precision floatint point format.
integer: an integer (positive or negative)
Class is metadata about the object that can determine how common functions operate on the object (ex: factor)
Factor: A factor is a vector that can contain only predefined values. It is used to store categorical data.

Types of Vectors

Logical, integer, double, character (list, NULL, complex)

```r
x <- factor(c("a", "b", "b", "a"))
x
```

```
[1] a b b a
Levels: a b
```

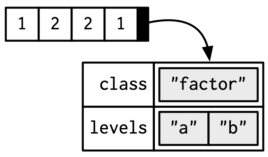
```r
typeof(x)
```

```
[1] "integer"
```

```r
attributes(x)
```

```
$levels
[1] "a" "b"

$class
[1] "factor"
```



Date:

```r
today <- Sys.Date()
today
```

```
[1] "2024-02-07"
```

```r
typeof(today)
```

```
[1] "double"
```

```r
attributes(today)
```

```
$class
[1] "Date"
```

logical (NA) → Int (1L, 1:3) → double (5) → character (“a”)