Intermediate R for Social Science

Miruna Barnoschi

October 30th, 2019

Contents

Overview

The goal of this intermediate R workshop is to start building the skills to (1) better utilize R Studio for social science research, (2) visualize data using the `ggplot2` package in R, (3) transform data using the `dplyr` package in R, and (4) use data visualization and transformation skills to explore social science data in a systematic way. The latter goals are first step into `tidyverse`, a suite of packages designed for data science that are immensely helpful when doing social science work.

Resources

There are several resources that would be helpful:
First, there is *R for Data Science* by Hadley Wickham and Garrett Grolemund, which is useful for learning to visualize, model, transform, tidy, and import data in a more formal way. It is free online: https://r4ds.had.co.nz.

Second, there is "stack overflow", which can be a lifesaver for any question that comes up with respect to R (see R tags): https://stackoverflow.com. If you have a question, there is someone who has probably asked it on stack overflow and got an answer.
Third, there is "r-bloggers", which can be helpful with respect to R explanations and tutorials: https://www.r-bloggers.com.
Fourth, there are the always handy R cheat sheets: https://rstudio.com/resources/cheatsheets/.
Fifth, there is of course google (a simple google search of the issue/error often does the trick).

R Packages

```
library(tinytex)
# package that provides LaTeX distribution for R:
# https://cran.r-project.org/web/packages/tinytex/tinytex.pdf
library(tidyverse)
# set of packages for data science:
# https://cran.r-project.org/web/packages/tidyverse/tidyverse.pdf
library(readr)
# package that provides a way to read data in R:
# https://cran.r-project.org/web/packages/readr/index.html
library(ggplot2)
# package that provides a way to create graphics:
# https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf
library(dplyr)
# package that provides a way to manipulate data:
# https://cran.r-project.org/web/packages/dplyr/dplyr.pdf
library(kableExtra)
# package to build complex HTML/LaTeX tables:
# https://cran.r-project.org/web/packages/kableExtra/kableExtra.pdf
library(gridExtra)
# package to arrange multiple grid-based plots on a page:
# https://cran.r-project.org/web/packages/gridExtra/gridExtra.pdf
```

R Project

One best practice organizationally is to create a new R Project in an existing directory (presumably in the folder where the data is saved and the research notes/paper using that data is to be saved). Doing so saves time in terms of setting a working directory every time one works on this research project; more importantly, doing so makes replication easier.

Dataset

We will look at a political science dataset that would be of interest to researchers in the subfields of comparative politics and international relations: "UCDP Battle-related Deaths Dataset Version 19.1".

Read Data into R

With the `readr` package, we can read in `BattleDeaths_v19_1.csv` using the appropriate function from the package, namely `read_csv()`. Datasets are usually "delimited" and `read_csv()` and `read_tsv()` are special cases of the general `read_delim()`, useful for reading comma separated values and tab separated values (i.e. the most common types of file data). There are other packages that can read data into R. With the `readxl` package, we can read in Excel fils (both .xls and .xlsx) using `read_excel()`. With the `haven` package, we can read in SPSS, Stata, and SAS files using `read_dta()`, `read_stata()`, `raed_sas()`, etc.

```
data <- read_csv("BattleDeaths_v19_1.csv")
```

Filtering, Selecting, and Arranging Data

The war in Afghanistan has gone on from 2001 till today. Let's examine more specifically battle deaths in Afghanistan from 2001 till today to get a better picture of the conflict in that location. We would do this by filtering the data by location, "Afghanistan", and by year (2001-present) and selecting the variables associated with battle deaths (`bd_best`, `bd_low`, `bd_high`).

```
data %>%
  filter(battle_location == "Afghanistan", year >= 2001) %>%
  arrange(year) %>%
  select(conflict_id, battle_location, year, bd_best, bd_low, bd_high)
```

```
## # A tibble: 14 x 6
##    conflict_id battle_location  year bd_best bd_low bd_high
##          <dbl> <chr>           <dbl>   <dbl>  <dbl>   <dbl>
##  1         333 Afghanistan      2001    2328   2217    7657
##  2       13692 Afghanistan      2001    1397   1388    2075
##  3         333 Afghanistan      2002      30     30      30
##  4         333 Afghanistan      2003      32     32      45
##  5         333 Afghanistan      2003     628    614     939
##  6         333 Afghanistan      2006      40     40      60
##  7         333 Afghanistan      2009      54     54      70
##  8         333 Afghanistan      2010      98     97     101
##  9         333 Afghanistan      2011      49     48      52
## 10         333 Afghanistan      2013      35     35      35
## 11       13637 Afghanistan      2015     674    625     865
## 12       13637 Afghanistan      2016    2141   2100    2544
## 13       13637 Afghanistan      2017    2795   2738    3009
## 14       13637 Afghanistan      2018    2842   2770    5149
```

We can change the R output of the data (tibble) to a more formatted table.

```
data %>%
  filter(battle_location == "Afghanistan", year >= 2001) %>%
  arrange(year) %>%
  select(conflict_id, battle_location, year, bd_best, bd_low, bd_high) %>%
  kable(col.names = c("Conflict ID",
                      "Battle Location",
                      "Year",
                      "Battle Deaths (best estimate)",
                      "Battle Deaths (low estimate)",
                      "Battle Deaths (high estimate)")) %>%
  kable_styling(latex_options="scale_down") # table is too big for page without this
```

Conflict ID	Battle Location	Year	Battle Deaths (best estimate)	Battle Deaths (low estimate)	Battle Deaths (high estimate)
333	Afghanistan	2001	2328	2217	7657
13692	Afghanistan	2001	1397	1388	2075
333	Afghanistan	2002	30	30	30
333	Afghanistan	2003	32	32	45
333	Afghanistan	2003	628	614	939
333	Afghanistan	2006	40	40	60
333	Afghanistan	2009	54	54	70
333	Afghanistan	2010	98	97	101
333	Afghanistan	2011	49	48	52
333	Afghanistan	2013	35	35	35
13637	Afghanistan	2015	674	625	865
13637	Afghanistan	2016	2141	2100	2544
13637	Afghanistan	2017	2795	2738	3009
13637	Afghanistan	2018	2842	2770	5149

Summary Statistics

We can find the summary statistics (e.g. mean) of battle deaths in the subset of the dataset that focuses on the time period of the war in Afghanistan. We can also do other calculations in R (e.g. addition/sum)

```
Afghanistandata <- data %>%
  filter(battle_location == "Afghanistan", year >= 2001) %>%
  arrange(year) %>%
  select(conflict_id, battle_location, year, bd_best, bd_low, bd_high)

summary(Afghanistandata$bd_best)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   30.00   42.25  363.00  938.79 1955.00 2842.00
```

```
summary(Afghanistandata$bd_low)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    30.0    42.0   355.5   913.4  1922.0  2770.0
```

```
summary(Afghanistandata$bd_high)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      30      54     483    1616    2427    7657
```

```
mean(Afghanistandata$bd_best)
```

```
## [1] 938.7857
```

```
mean(Afghanistandata$bd_low)
```

```
## [1] 913.4286
```

```
mean(Afghanistandata$bd_high)
```

```
## [1] 1616.5
```

```
sum(Afghanistandata$bd_best)
```

```
## [1] 13143
```

```
sum(Afghanistandata$bd_low)
```

```
## [1] 12788
```

```
sum(Afghanistandata$bd_high)
```

```
## [1] 22631
```

As can be seen from the table displaying the Afghanistan subset of the data corresponding to the war in Afghanistan time period, there were three distinct conflicts at that time. Should that matter for our battle deaths calculations? It is always important to know the exact data that should be analyzed prior to doing any analysis.

Do it yourself: How would you examine battle deaths in the war in Iraq, which lasted from 2003 till 2011?

Finding Data

Let's assume we do not want to limit ourselves to looking at data during the war in Afghanistan, but instead at data that involves Afghanistan (regardless if battles deaths occurred in Afghanistan per se). We could do so by filtering each variable with "Afghanistan". Or, more simply, we could filter all the data for any variables with the string pattern, "Afghanistan". There are numerous handy shortcuts in `dplyr` that can help us identify and find subsets of the data/specific values and information in the data.

```
data %>%
  filter_all(any_vars(str_detect(., pattern = "Afghanistan"))) %>%
  arrange(year) %>%
  select(conflict_id, location_inc, side_a, side_a_2nd, side_b, side_b_2nd, year, battle_location, conta
```

```
## # A tibble: 90 x 11
##    conflict_id location_inc side_a side_a_2nd side_b side_b_2nd  year
##          <dbl> <chr>        <chr>  <chr>      <chr>  <chr>      <dbl>
##  1         333 Afghanistan  Gover~ Governmen~ Jam'i~ <NA>        1989
##  2         333 Afghanistan  Gover~ Governmen~ Hizb-~ <NA>        1989
##  3         333 Afghanistan  Gover~ Governmen~ Mahaz~ <NA>        1989
##  4         333 Afghanistan  Gover~ Governmen~ Hizb-~ <NA>        1989
##  5         333 Afghanistan  Gover~ Governmen~ Hizb-~ <NA>        1989
##  6         333 Afghanistan  Gover~ <NA>       Jam'i~ <NA>        1990
##  7         333 Afghanistan  Gover~ <NA>       Hizb-~ <NA>        1990
##  8         333 Afghanistan  Gover~ <NA>       Hizb-~ <NA>        1990
##  9         333 Afghanistan  Gover~ <NA>       Hizb-~ <NA>        1990
## 10         333 Afghanistan  Gover~ <NA>       Milit~ <NA>        1990
## # ... with 80 more rows, and 4 more variables: battle_location <chr>,
## #   bd_best <dbl>, bd_low <dbl>, bd_high <dbl>
```

Grouping Data

Suppose we wanted to know where and when the most amount of battle deaths occured. We could arrange the data to show the most battle death occurrences first. But, if we wanted to know where and when the most battle death occurred (in the aggregate), we should group by region/battle location and year and then tally up the battle deaths to see which region/battle location and year have the most battle death occurrences.

```
data %>%
  arrange(desc(bd_best)) %>%
  select(conflict_id, battle_location, year, bd_best)
```

```
## # A tibble: 1,565 x 4
##    conflict_id battle_location                          year bd_best
```

```
##          <dbl> <chr>                                    <dbl>   <dbl>
##  1         299 Iraq, Lebanon, Syria                      2013   68614
##  2         299 Lebanon, Syria                            2014   54547
##  3         409 Eritrea, Ethiopia                         2000   50000
##  4         409 Eritrea, Ethiopia                         1999   47192
##  5         299 Syria, Turkey                             2012   39939
##  6         299 Jordan, Syria                             2015   33956
##  7         275 Ethiopia                                  1990   30633
##  8         299 Syria                                     2016   27207
##  9         333 Afghanistan, Pakistan                     2018   22837
## 10         371 Iraq, Kuwait, Philippines, Saudi Arabia   1991   21790
## # ... with 1,555 more rows
```

```
data %>%
  group_by(region) %>%
  tally(bd_best, sort = TRUE)
```

```
## # A tibble: 6 x 2
##   region      n
##    <dbl>  <dbl>
## 1      4 458105
## 2      2 444290
## 3      3 397851
## 4      1  58848
## 5      5  40464
## 6     NA   9324
```

```
data %>%
  group_by(battle_location) %>%
  tally(bd_best, sort = TRUE)
```

```
## # A tibble: 171 x 2
##    battle_location            n
##    <chr>                  <dbl>
##  1 Afghanistan, Pakistan 150402
##  2 Eritrea, Ethiopia      98192
##  3 Ethiopia               88036
##  4 Syria                  77440
##  5 Iraq, Lebanon, Syria   68614
##  6 Afghanistan            66970
##  7 Lebanon, Syria         63974
##  8 Sri Lanka              57366
##  9 Syria, Turkey          40571
## 10 Sudan                  37398
## # ... with 161 more rows
```

```
data %>%
  group_by(year) %>%
  tally(bd_best, sort = TRUE)
```

```
## # A tibble: 30 x 2
##     year      n
##    <dbl>  <dbl>
##  1  2014 104555
##  2  2015  97328
##  3  2013  90596
```

```
##  4  2016  85799
##  5  1999  80578
##  6  1990  79681
##  7  2000  77357
##  8  1991  70236
##  9  2017  67101
## 10  2012  63257
## # ... with 20 more rows
```

```
data %>%
  group_by(region, year) %>%
  tally(bd_best, sort = TRUE)
```

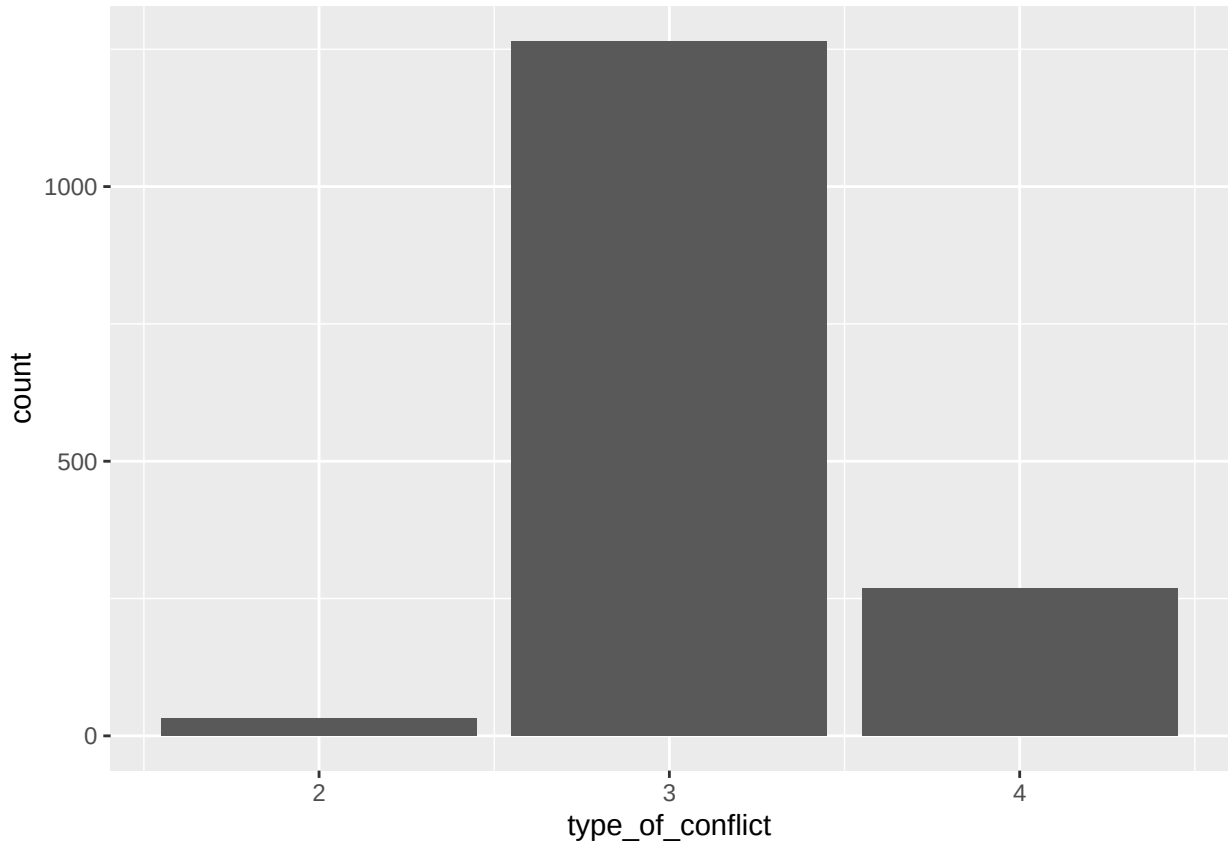
```
## # A tibble: 152 x 3
## # Groups:   region [6]
##    region  year     n
##     <dbl> <dbl> <dbl>
##  1      2  2014 73051
##  2      2  2013 71684
##  3      2  2015 64595
##  4      4  1990 64585
##  5      4  1999 61216
##  6      4  2000 59700
##  7      2  2016 55376
##  8      2  2012 43734
##  9      2  2017 34621
## 10      4  1989 34546
## # ... with 142 more rows
```

Why do each of these tables give a different intuition about where and when the most amount of battle deaths occured? It is always important to keep in mind which variables we are using and how we are using those variables.

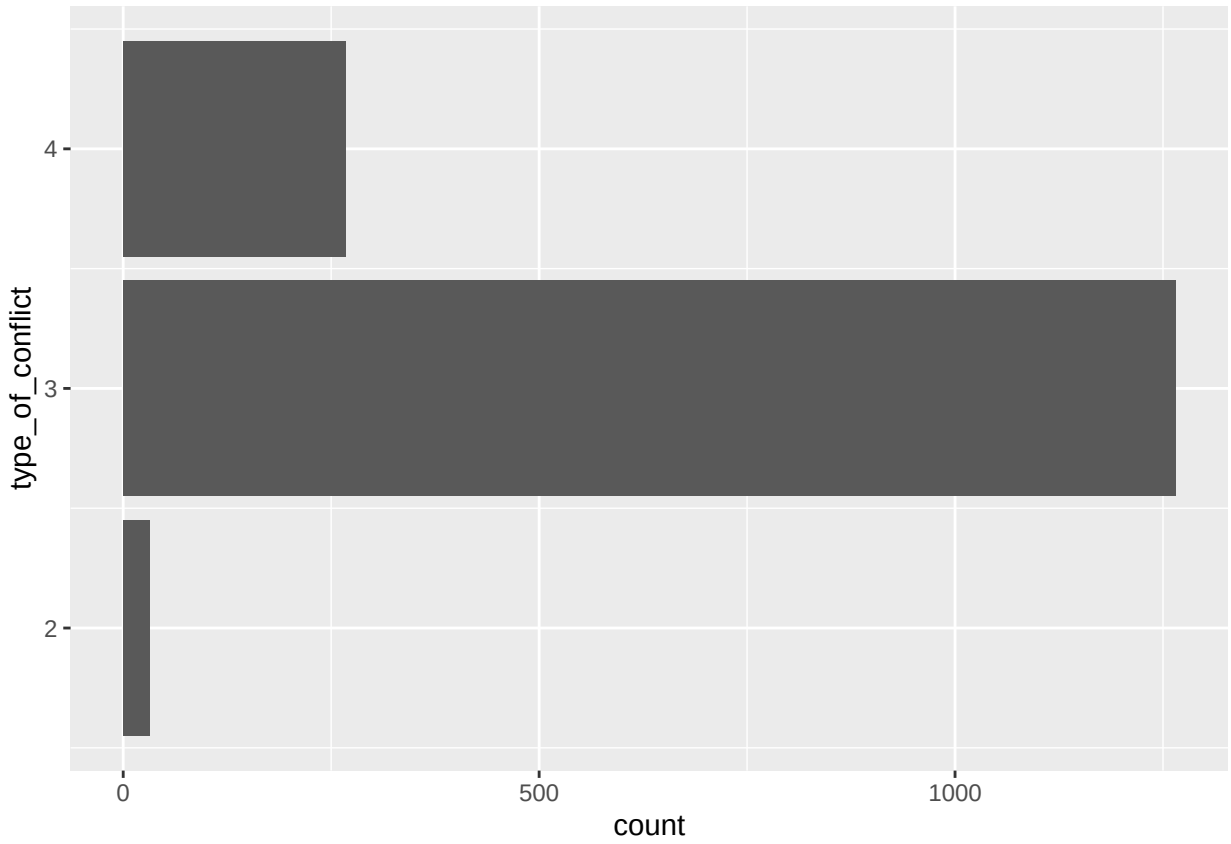
Bar Charts

Suppose we wanted to know the distribution of the types of conflict this dataset contains (i.e. the distribution of the types of conflict that occurred during 1989-2018). We can create a bar chart to visualize this distribution.

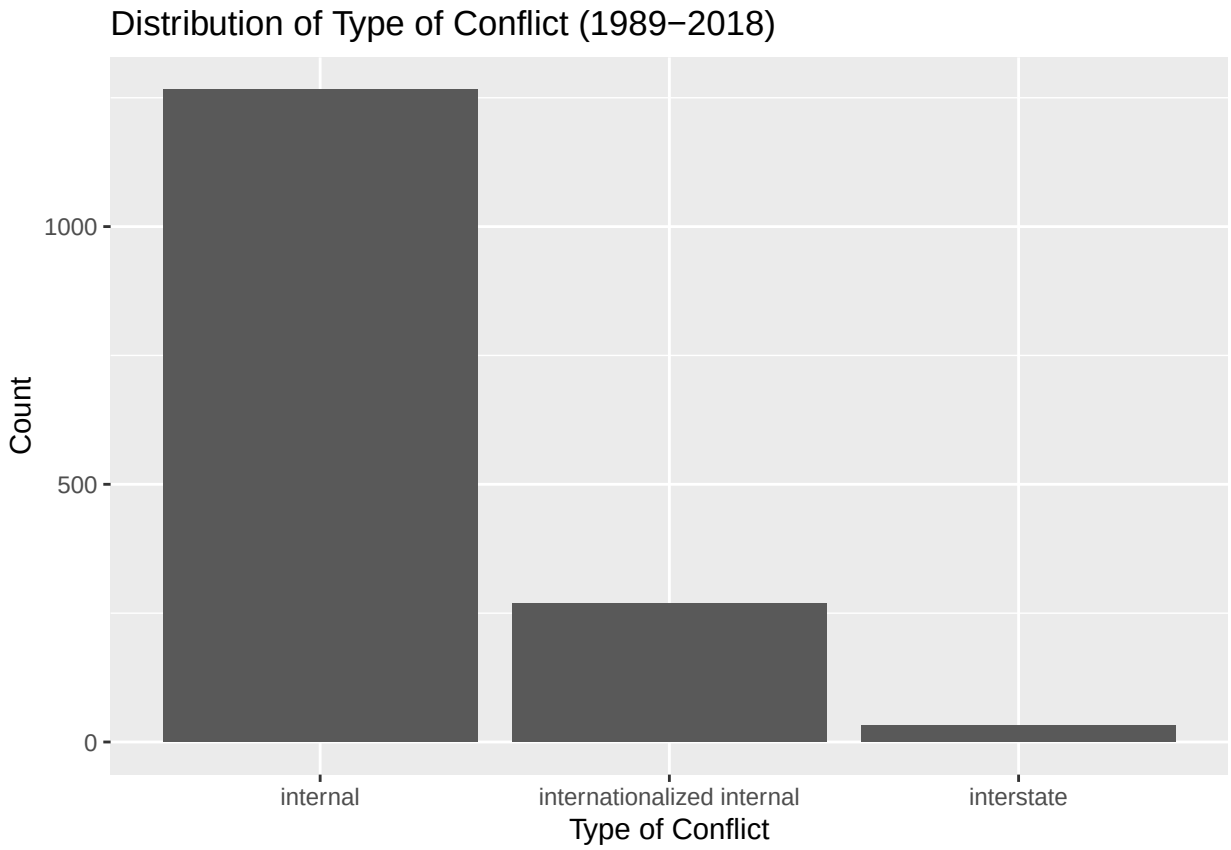
```
data %>%
  ggplot(mapping = aes(x = type_of_conflict)) +
  geom_bar()
```




```
data %>%
  ggplot(mapping = aes(x = type_of_conflict)) +
  geom_bar() +
  coord_flip()
```



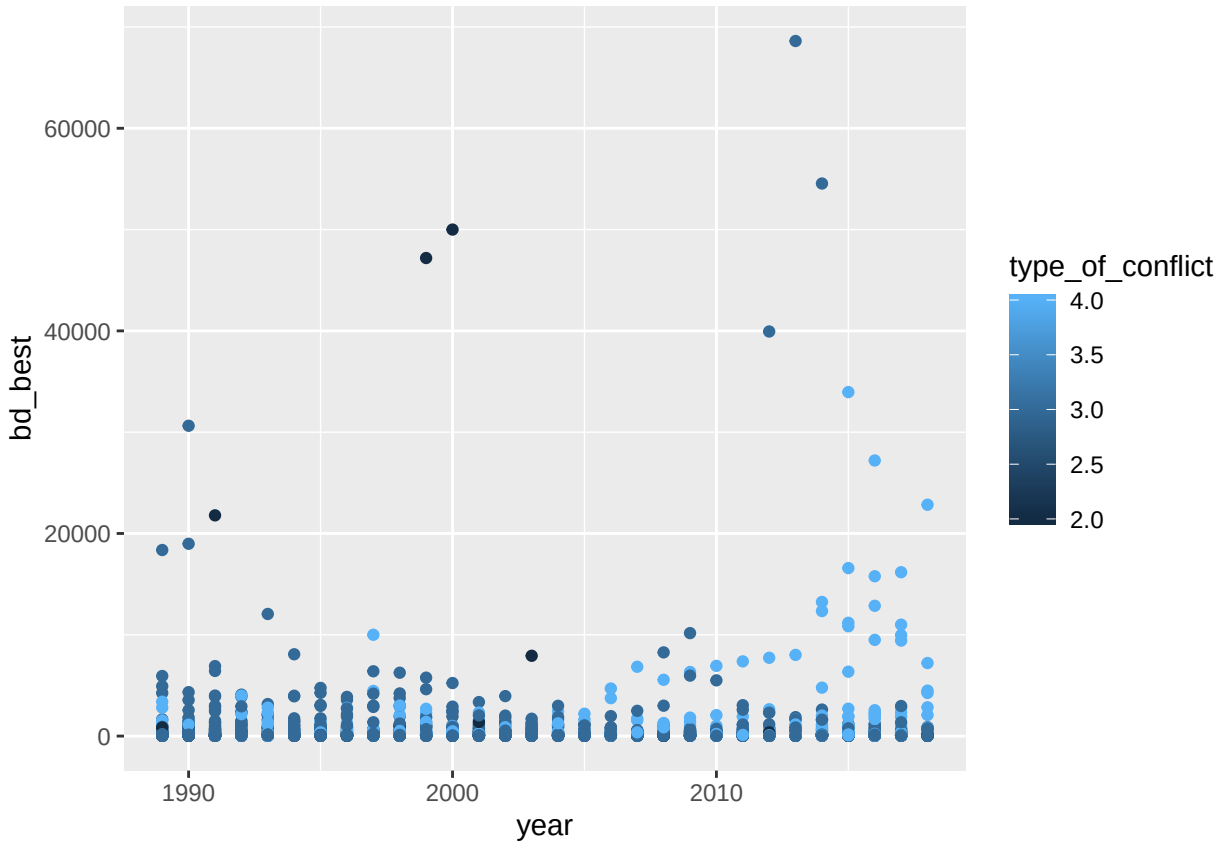

```
data %>%
  mutate(type_of_conflict = case_when(
    type_of_conflict == "2" ~ "interstate",
    type_of_conflict == "3" ~ "internal",
    type_of_conflict == "4" ~ "internationalized internal")) %>%
  ggplot(mapping = aes(x = type_of_conflict)) +
  geom_bar() +
  ggtitle("Distribution of Type of Conflict (1989-2018)") +
  xlab("Type of Conflict") + ylab("Count")
```



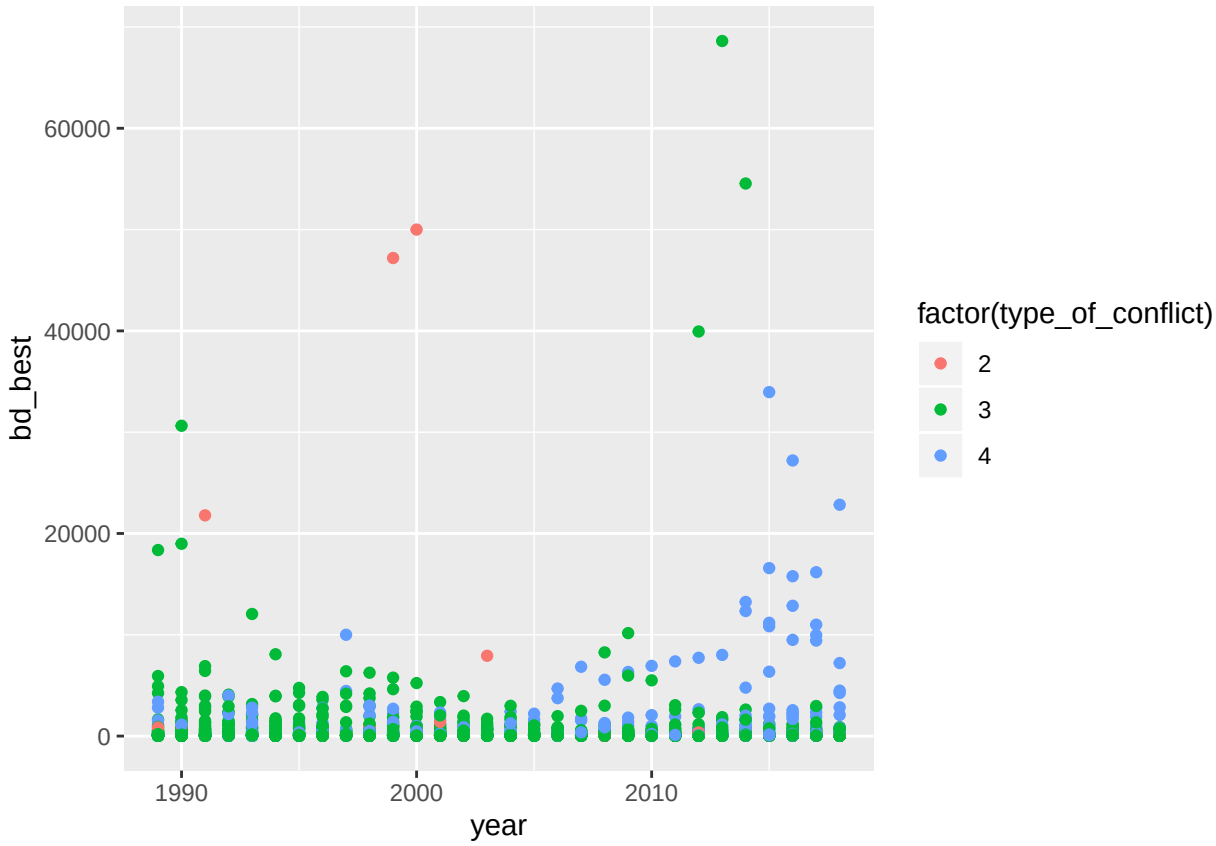
Scatter Plots

Suppose we wanted to see the number of battle deaths across the time period of the dataset (1989-2018) and identify whether there is a pattern. We can create a scatter plot to visualize this. Indeed, with a scatter plot, we can even subset the points by type of conflict, adding another layer of information about the pattern.

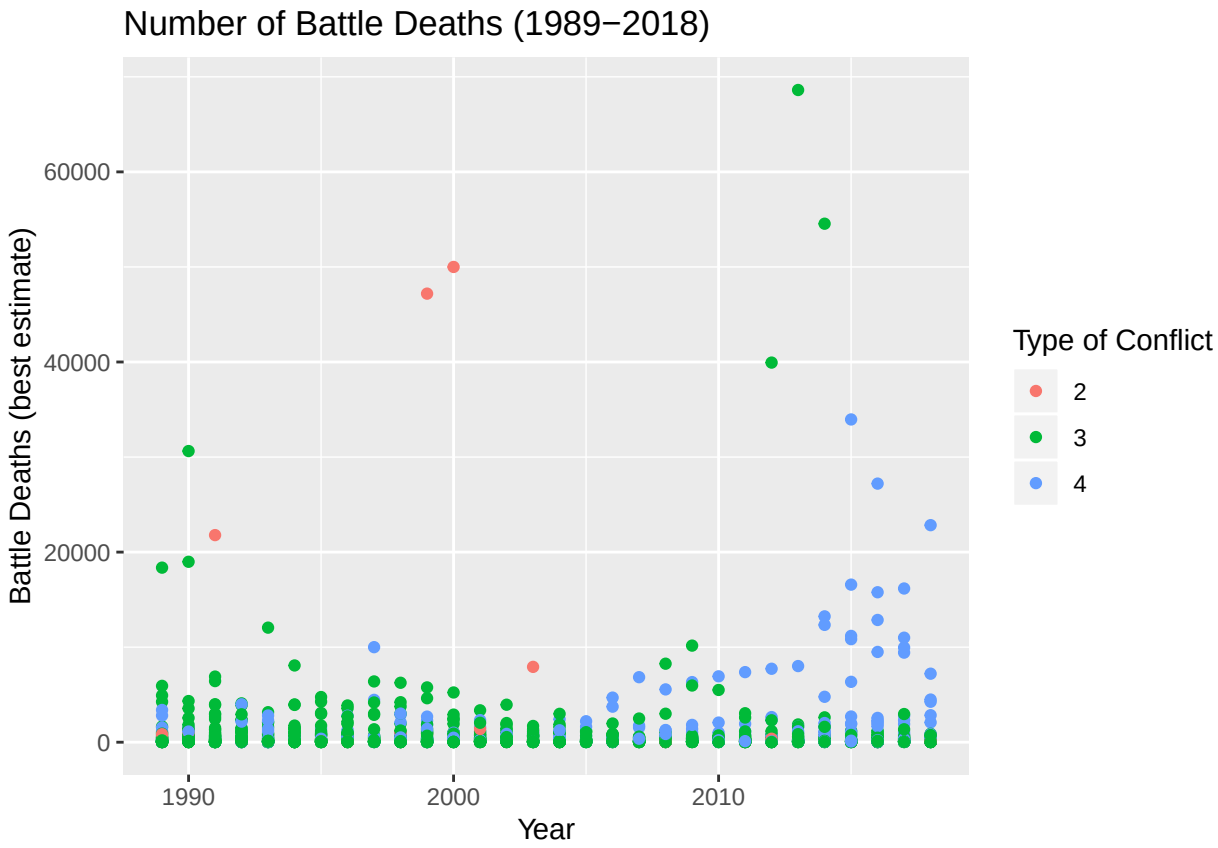
```
data %>%
  ggplot(mapping = aes(x = year, y = bd_best, color = type_of_conflict)) +
  geom_point()
```




```
# Factor `type_of_conflict`

data %>%
  ggplot(mapping = aes(x = year, y = bd_best, colour = factor(type_of_conflict))) +
  geom_point()
```

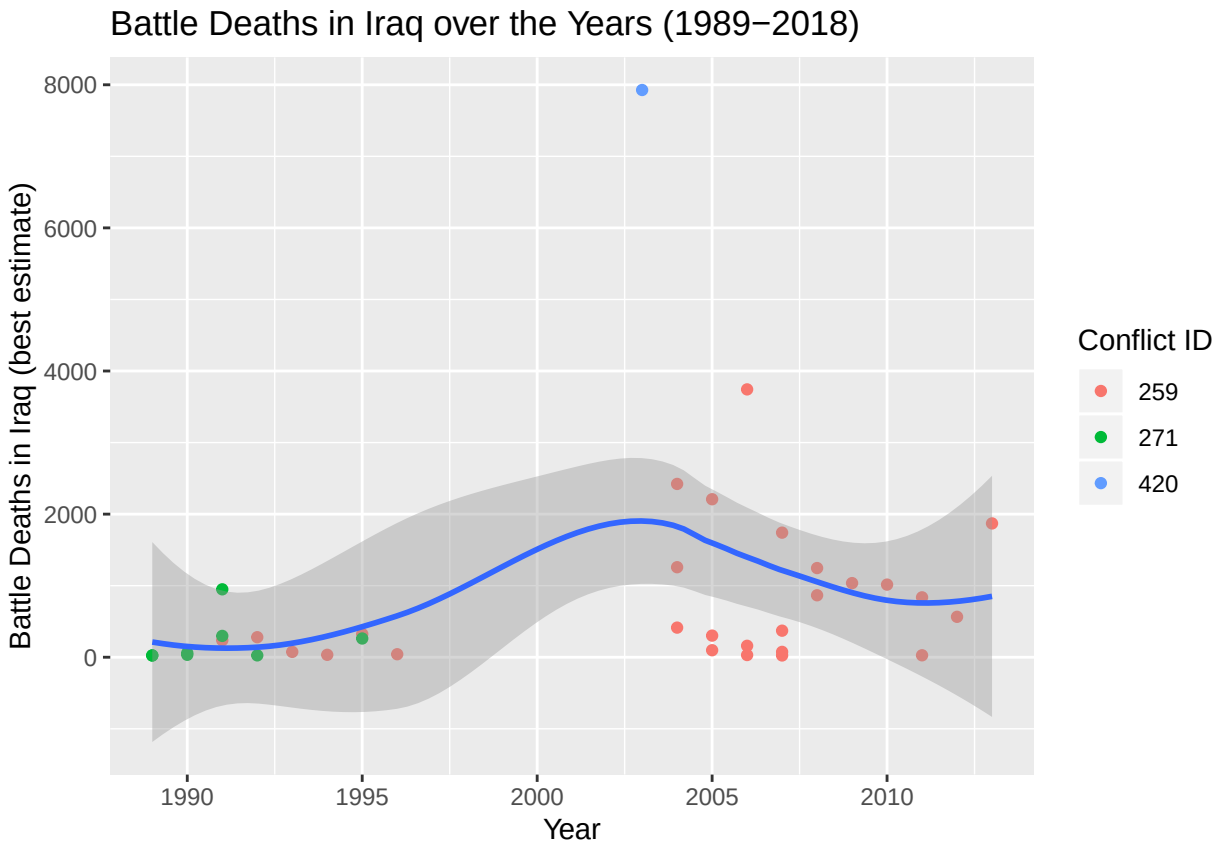



```
data %>%
  ggplot(mapping = aes(x = year, y = bd_best, color = factor(type_of_conflict))) +
  geom_point() +
  ggtitle("Number of Battle Deaths (1989-2018)") +
  labs(x = "Year", y = "Battle Deaths (best estimate)", color = "Type of Conflict")
```



Suppose we wanted to know the number of battle deaths across time in Iraq and determine the trend. We can create a scatterplot showing the battle deaths across time for Iraq (even subset the points by type of conflict) and add a trend line.

```
data %>%
  filter(battle_location == "Iraq") %>%
  arrange(conflict_id, year) %>%
  select(conflict_id, battle_location, year, bd_best) %>%
  ggplot(mapping = aes(x = year, y = bd_best)) +
  geom_point(aes(color = factor(conflict_id))) +
  geom_smooth() +
  ggtitle("Battle Deaths in Iraq over the Years (1989-2018)") +
  labs(x = "Year", y = "Battle Deaths in Iraq (best estimate)", colour = "Conflict ID")
```



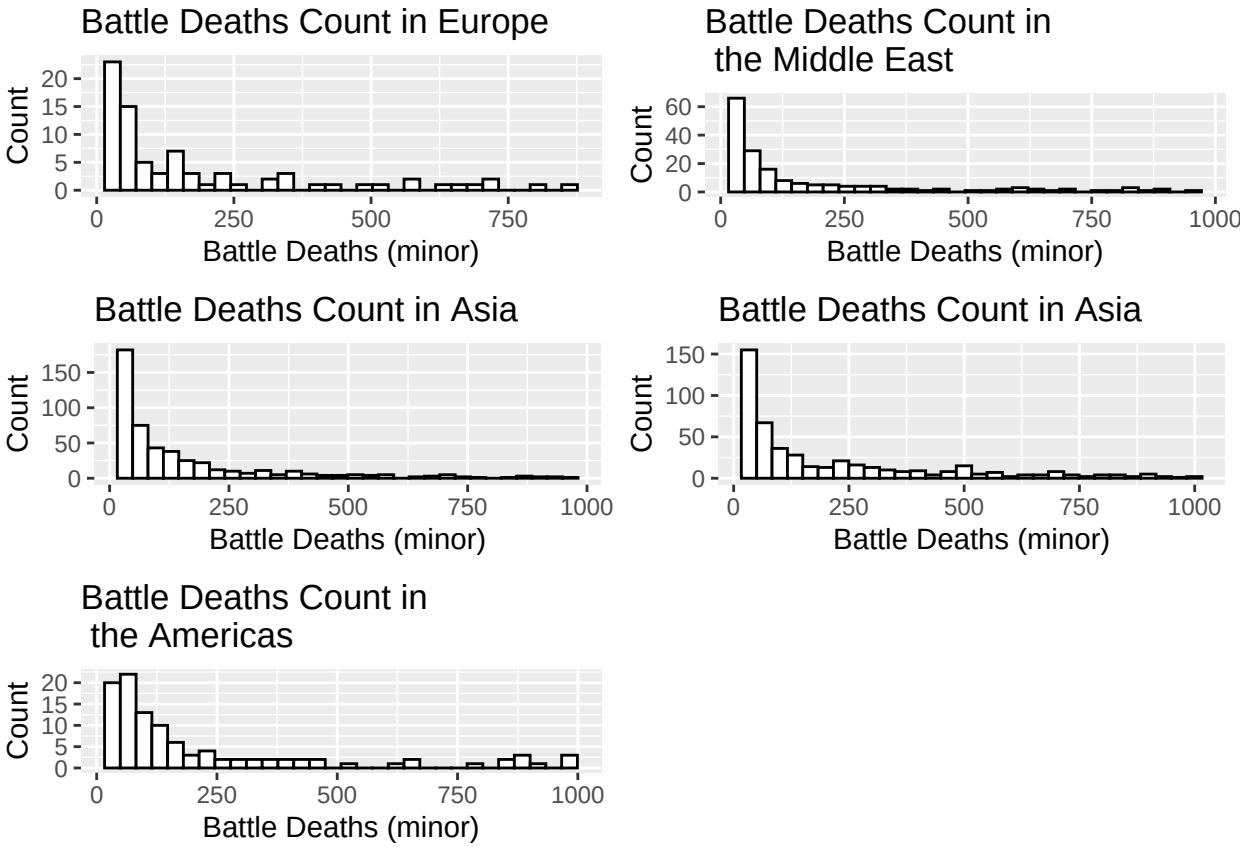
Do it yourself: How would you see the number of battle deaths across the time period of 1989-2018 and identify whether there is a pattern with respect to the main conflict issue? (hint: `incompatibility`)

Histograms

Suppose we wanted to visually summarize the number of times battle deaths of minor intensity occurred in any given year (between 25 and 999 battle-related deaths) and see the shape and spread of the freqeuncy of these battle death numbers for each region. We can create histograms that show the number of times battle deaths of minor intensity occurred for each region.

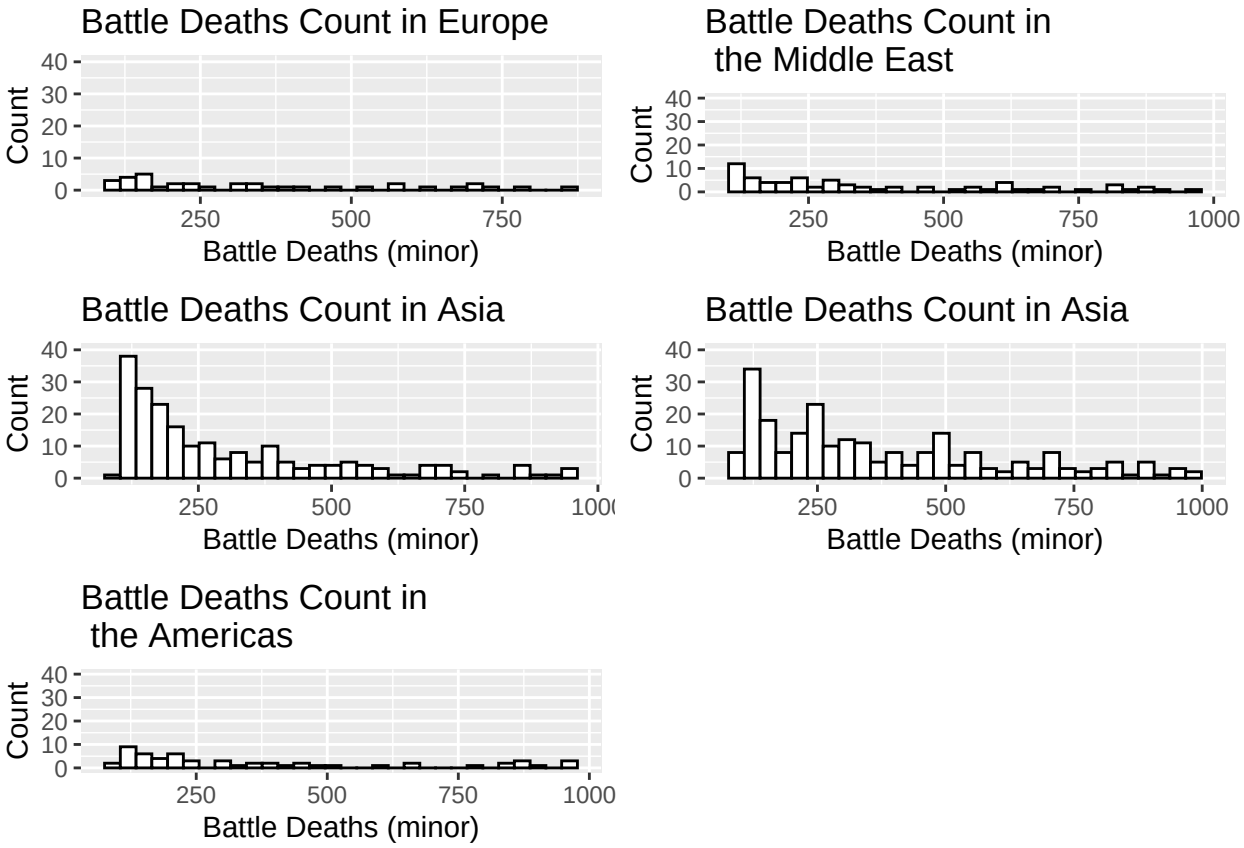
```
a1 <- data %>%
  filter(bd_best <= 999, region == 1) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  ggtitle("Battle Deaths Count in Europe") +
  labs(x = "Battle Deaths (minor)", y = "Count")

a2 <-  data %>%
  filter(bd_best <= 999, region == 2) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  ggtitle("Battle Deaths Count in \n the Middle East") +
  labs(x = "Battle Deaths (minor)", y = "Count")
# \n is for new line (R Studio special characters)

a3 <- data %>%
```

```
  filter(bd_best <= 999, region == 3) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  ggtitle("Battle Deaths Count in Asia") +
  labs(x = "Battle Deaths (minor)", y = "Count")

a4 <- data %>%
  filter(bd_best <= 999, region == 4) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  ggtitle("Battle Deaths Count in Asia") +
  labs(x = "Battle Deaths (minor)", y = "Count")

a5 <- data %>%
  filter(bd_best <= 999, region == 5) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  ggtitle("Battle Deaths Count in \n the Americas") +
  labs(x = "Battle Deaths (minor)", y = "Count")
# \n is for new line (R Studio special characters)

grid.arrange(a1, a2,
             a3, a4,
             a5)
```




```
# Comparison between the regions proves difficult because of y axis scaling
```

```r
b1 <- data %>%
  filter(bd_best <= 999, bd_best >=100, region == 1) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  coord_cartesian(ylim=c(0,40)) +
  ggtitle("Battle Deaths Count in Europe") +
  labs(x = "Battle Deaths (minor)", y = "Count")

b2 <-  data %>%
  filter(bd_best <= 999, bd_best >=100, region == 2) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  coord_cartesian(ylim=c(0,40)) +
  ggtitle("Battle Deaths Count in \n the Middle East") +
  labs(x = "Battle Deaths (minor)", y = "Count")

b3 <- data %>%
  filter(bd_best <= 999, bd_best >=100, region == 3) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  coord_cartesian(ylim=c(0,40)) +
  ggtitle("Battle Deaths Count in Asia") +
  labs(x = "Battle Deaths (minor)", y = "Count")

b4 <- data %>%
  filter(bd_best <= 999, bd_best >=100, region == 4) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  coord_cartesian(ylim=c(0,40)) +
  ggtitle("Battle Deaths Count in Asia") +
  labs(x = "Battle Deaths (minor)", y = "Count")

b5 <- data %>%
  filter(bd_best <= 999, bd_best >=100, region == 5) %>%
  ggplot(mapping = aes(x = bd_best)) +
  geom_histogram(color = "black", fill="white") +
  coord_cartesian(ylim=c(0,40)) +
  ggtitle("Battle Deaths Count in \n the Americas") +
  labs(x = "Battle Deaths (minor)", y = "Count")

grid.arrange(b1, b2,
             b3, b4,
             b5)
```

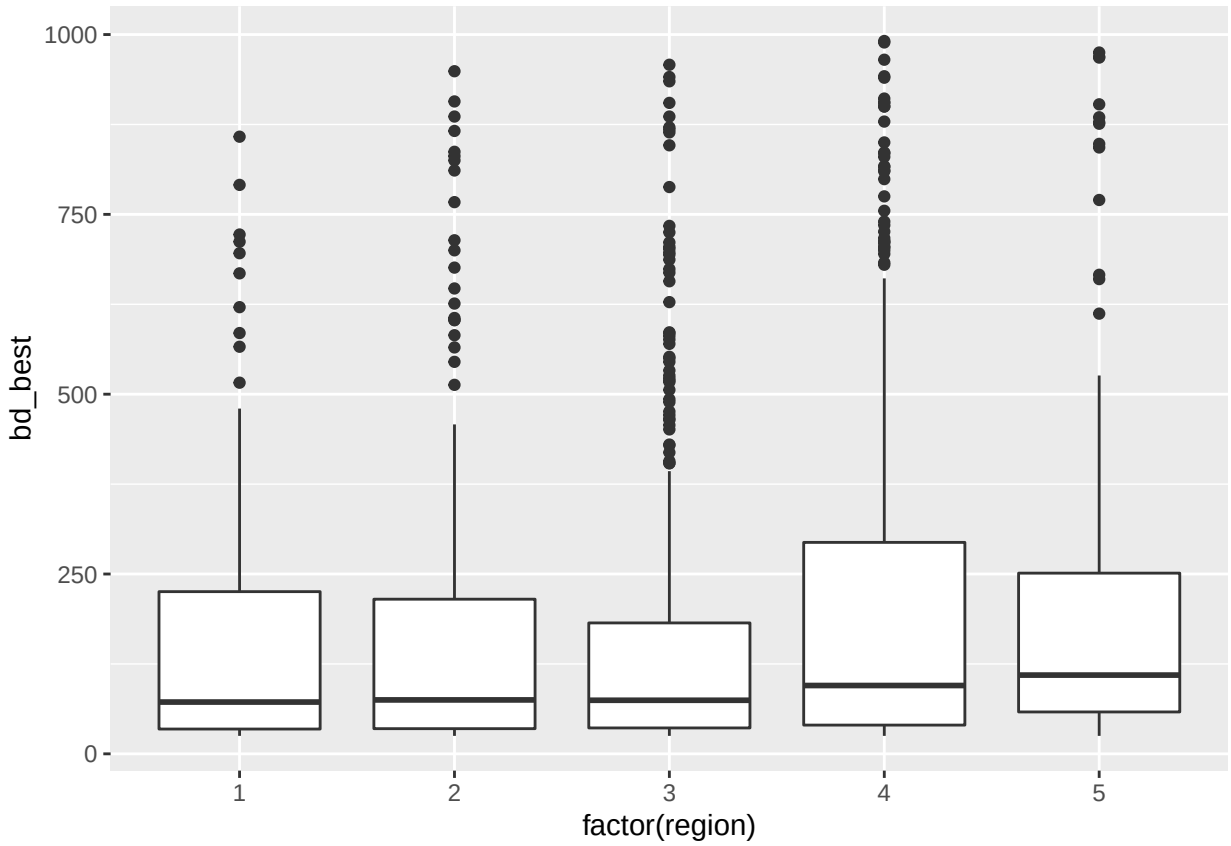


Do it yourself: How would you visually summarize the number of times battle deaths of major intensity occurred in any given year (greater than 999 battle-related deaths) and see the shape and spread of the freqeuncy of these battle death numbers for each region?

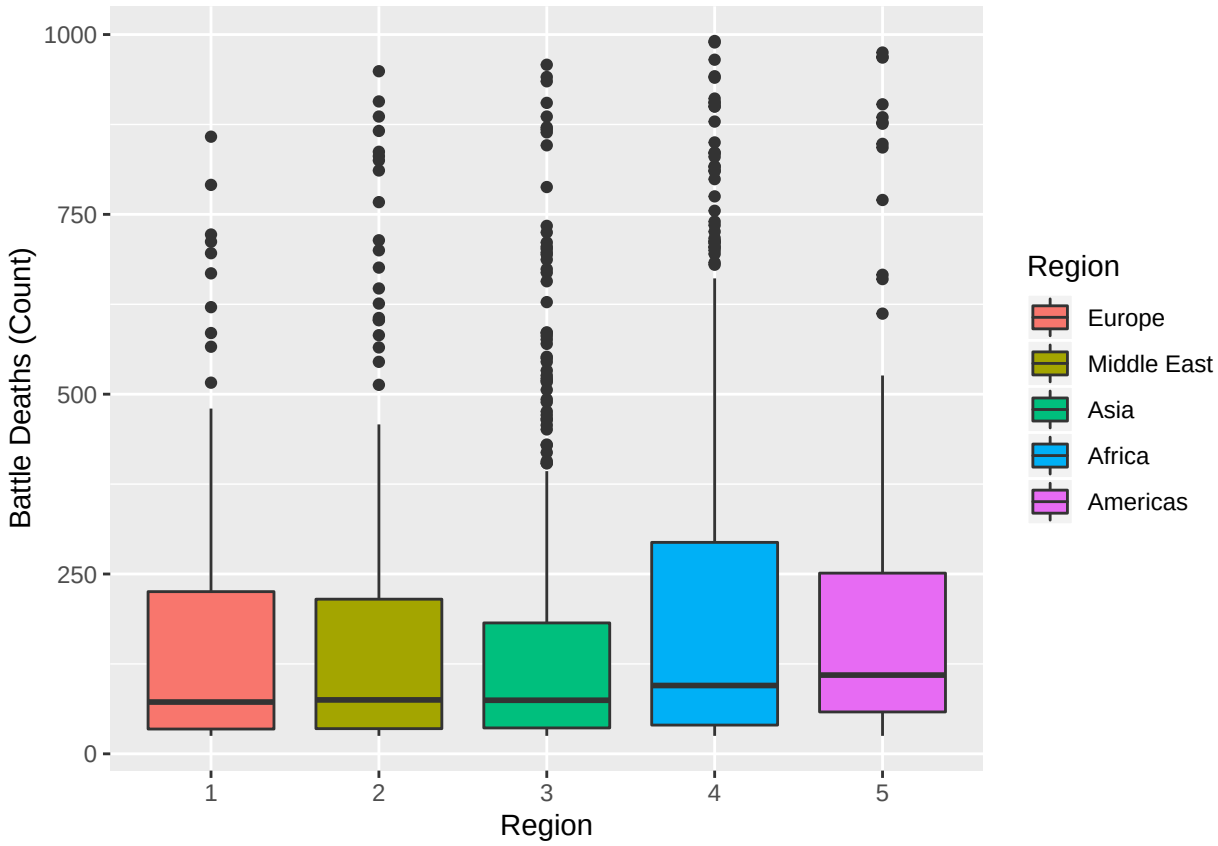
Box Plots

Suppose we wanted to visually summarize the number of times battle deaths of minor intensity occurred in any given year for each region in a way that would allow for better comparison than the previous histograms. We can create side-by-side boxplots.

```
data %>%
  filter(bd_best <= 999, region) %>%
  ggplot(mapping = aes(x = factor(region), y = bd_best)) +
  geom_boxplot()
```




```
data %>%
  filter(bd_best <= 999, region) %>%
  ggplot(mapping = aes(x = factor(region), y = bd_best, fill = factor(region))) +
  geom_boxplot() +
  labs(x = "Region", y = "Battle Deaths (Count)") +
  scale_fill_discrete(name = "Region", labels = c("Europe",
                                                  "Middle East",
                                                  "Asia",
                                                  "Africa",
                                                  "Americas"))
```



Next Steps

In order to do more advanced modeling and analysis of the data (that would allow for hypothesis testing), one should go deeper into understanding and fixing problems with the dataset (e.g. missing values or NA's), tidying the data, and learnng the R tools that would be best suited for that data (be it `lm()`, `glm()`, `nnet()`, etc.).