Mid2-reviw

Zhuoya Wang

2024-02-26

#install.packages("naniar")  
library("tidyverse")

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library("naniar")  
library("skimr")

##   
## Attaching package: 'skimr'  
##   
## The following object is masked from 'package:naniar':  
##   
## n\_complete

library("janitor")

##   
## Attaching package: 'janitor'  
##   
## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

# Lab 8 - NA’s

1. Define NA and describe how they are treated in R.
2. Produce summaries of the number of NA’s in a data set.
3. Replace values with NA in a data set as appropriate.

life\_history <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab8/data/mammal\_lifehistories\_v3.csv") %>% clean\_names()

## Rows: 1440 Columns: 13  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): order, family, Genus, species, newborn  
## dbl (8): mass, gestation, weaning, wean mass, AFR, max. life, litter size, l...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

### Are there any NA’s

# Check NAs  
  
summary(life\_history)# check if there are any wired values(eg. -999)

## order family genus species   
## Length:1440 Length:1440 Length:1440 Length:1440   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## mass gestation newborn weaning   
## Min. : -999 Min. :-999.00 Length:1440 Min. :-999.00   
## 1st Qu.: 50 1st Qu.:-999.00 Class :character 1st Qu.:-999.00   
## Median : 403 Median : 1.05 Mode :character Median : 0.73   
## Mean : 383577 Mean :-287.25 Mean :-427.17   
## 3rd Qu.: 7009 3rd Qu.: 4.50 3rd Qu.: 2.00   
## Max. :149000000 Max. : 21.46 Max. : 48.00   
##   
## wean\_mass afr max\_life litter\_size   
## Min. : -999 Min. :-999.00 Min. : 0.00 Min. :-999.000   
## 1st Qu.: -999 1st Qu.:-999.00 1st Qu.: 0.00 1st Qu.: 1.000   
## Median : -999 Median : 2.50 Median : 0.00 Median : 2.270   
## Mean : 16049 Mean :-408.12 Mean : 93.19 Mean : -55.634   
## 3rd Qu.: 10 3rd Qu.: 15.61 3rd Qu.: 147.25 3rd Qu.: 3.835   
## Max. :19075000 Max. : 210.00 Max. :1368.00 Max. : 14.180   
##   
## litters\_year   
## Min. :0.140   
## 1st Qu.:1.000   
## Median :1.000   
## Mean :1.636   
## 3rd Qu.:2.000   
## Max. :7.500   
## NA's :689

life\_history %>% map\_df(~ sum(is.na(.))) #This will give you a quick summary of the number of NA's in each variable.

## # A tibble: 1 × 13  
## order family genus species mass gestation newborn weaning wean\_mass afr  
## <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>  
## 1 0 0 0 0 0 0 0 0 0 0  
## # ℹ 3 more variables: max\_life <int>, litter\_size <int>, litters\_year <int>

This will give you a quick summary of the number of NA’s in each variable. Notice that, at least for now, it doesn’t look like there are any NA’s. But, given the size of the data this is unlikely.

### creating a new dataframe w/out NA

life\_history%>%  
 replace\_with\_na(replace = list(x= "-999", "-999.00", "NA", " ", ".", "not measured"))%>%  
 clean\_names()

## Warning in replace\_with\_na.data.frame(., replace = list(x = "-999", "-999.00",  
## : Missing from data: x,

## # A tibble: 1,440 × 13  
## order family genus species mass gestation newborn weaning wean\_mass afr  
## <chr> <chr> <chr> <chr> <dbl> <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 Artio… Antil… Anti… americ… 4.54e4 8.13 3246.36 3 8900 13.5  
## 2 Artio… Bovid… Addax nasoma… 1.82e5 9.39 5480 6.5 -999 27.3  
## 3 Artio… Bovid… Aepy… melamp… 4.15e4 6.35 5093 5.63 15900 16.7  
## 4 Artio… Bovid… Alce… busela… 1.5 e5 7.9 10166.… 6.5 -999 23.0  
## 5 Artio… Bovid… Ammo… clarkei 2.85e4 6.8 not me… -999 -999 -999   
## 6 Artio… Bovid… Ammo… lervia 5.55e4 5.08 3810 4 -999 14.9  
## 7 Artio… Bovid… Anti… marsup… 3 e4 5.72 3910 4.04 -999 10.2  
## 8 Artio… Bovid… Anti… cervic… 3.75e4 5.5 3846 2.13 -999 20.1  
## 9 Artio… Bovid… Bison bison 4.98e5 8.93 20000 10.7 157500 29.4  
## 10 Artio… Bovid… Bison bonasus 5 e5 9.14 23000.… 6.6 -999 30.0  
## # ℹ 1,430 more rows  
## # ℹ 3 more variables: max\_life <dbl>, litter\_size <dbl>, litters\_year <dbl>

life\_history %>% map\_df(~ sum(is.na(.))) # recheck

## # A tibble: 1 × 13  
## order family genus species mass gestation newborn weaning wean\_mass afr  
## <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>  
## 1 0 0 0 0 0 0 0 0 0 0  
## # ℹ 3 more variables: max\_life <int>, litter\_size <int>, litters\_year <int>

# dataset%>%  
 # replace\_with\_na\_all(condition = ~.x == -999)   
# clean data like -999.0, -999.00, -999.000 ....

## miss\_var\_summary provides a clean summary of NA’s across the data frame.

naniar::miss\_var\_summary(life\_history)

## # A tibble: 13 × 3  
## variable n\_miss pct\_miss  
## <chr> <int> <dbl>  
## 1 litters\_year 689 47.8  
## 2 order 0 0   
## 3 family 0 0   
## 4 genus 0 0   
## 5 species 0 0   
## 6 mass 0 0   
## 7 gestation 0 0   
## 8 newborn 0 0   
## 9 weaning 0 0   
## 10 wean\_mass 0 0   
## 11 afr 0 0   
## 12 max\_life 0 0   
## 13 litter\_size 0 0

# percent of missing and how many missing values  
  
life\_history <-   
 life\_history %>%   
 mutate(max\_life=na\_if(max\_life, 0)) # replace 0 with NA in col called max\_life  
  
miss\_var\_summary(life\_history) #check again

## # A tibble: 13 × 3  
## variable n\_miss pct\_miss  
## <chr> <int> <dbl>  
## 1 max\_life 841 58.4  
## 2 litters\_year 689 47.8  
## 3 order 0 0   
## 4 family 0 0   
## 5 genus 0 0   
## 6 species 0 0   
## 7 mass 0 0   
## 8 gestation 0 0   
## 9 newborn 0 0   
## 10 weaning 0 0   
## 11 wean\_mass 0 0   
## 12 afr 0 0   
## 13 litter\_size 0 0

## We can also use `miss\_var\_summary` with `group\_by()`. This helps us better evaluate where NA's are in the data.  
  
#life\_history %>%  
# group\_by(order) %>%  
# select(order, wean\_mass) %>%   
# miss\_var\_summary(order=T)

If you are sure that you know how NA’s are treated in the data, then you can deal with them in advance using na() as part of the readr package.

#life\_history\_advance <-   
# readr::read\_csv(file = "data/mammal\_lifehistories\_v3.csv",   
# na = c("NA", " ", ".", "-999")) #all NA, blank spaces, .,and -999 are treated as NA

# Lab 9.1 Tidy data and pivot\_long()Tidy` data follows three conventions:

1. each variable has its own column
2. each observation has its own row
3. each value has its own cell

#### When the wide format creates a problem because column names may actually represent values of a variable. The command pivot\_longer() shifts data from wide to long format.

### Rules:

* pivot\_longer(cols, names\_to, values\_to)
* cols - Columns to pivot to longer format
* names\_to - Name of the new column; it will contain the column names of gathered columns as values
* values\_to - Name of the new column; it will contain the data stored in the values of gathered columns

library(tidyverse)  
heartrate <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab9/data/heartrate.csv")

## Rows: 6 Columns: 5  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): patient  
## dbl (4): a, b, c, d  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

heartrate

## # A tibble: 6 × 5  
## patient a b c d  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Margaret 72 74 80 68  
## 2 Frank 84 84 88 76  
## 3 Hawkeye 64 66 68 64  
## 4 Trapper 60 58 64 58  
## 5 Radar 74 72 78 70  
## 6 Henry 88 87 88 72

heartrate%>%   
 pivot\_longer(-patient, #patient will not move  
 names\_to = "drug", #make a new column called "drug"  
 values\_to="heartrate" #values moved to a new column called heartrate  
 )

## # A tibble: 24 × 3  
## patient drug heartrate  
## <chr> <chr> <dbl>  
## 1 Margaret a 72  
## 2 Margaret b 74  
## 3 Margaret c 80  
## 4 Margaret d 68  
## 5 Frank a 84  
## 6 Frank b 84  
## 7 Frank c 88  
## 8 Frank d 76  
## 9 Hawkeye a 64  
## 10 Hawkeye b 66  
## # ℹ 14 more rows

relig\_income <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab9/data/relig\_income.csv");relig\_income

## Rows: 18 Columns: 11  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): religion  
## dbl (10): <$10k, $10-20k, $20-30k, $30-40k, $40-50k, $50-75k, $75-100k, $100...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## # A tibble: 18 × 11  
## religion `<$10k` `$10-20k` `$20-30k` `$30-40k` `$40-50k` `$50-75k` `$75-100k`  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Agnostic 27 34 60 81 76 137 122  
## 2 Atheist 12 27 37 52 35 70 73  
## 3 Buddhist 27 21 30 34 33 58 62  
## 4 Catholic 418 617 732 670 638 1116 949  
## 5 Don’t k… 15 14 15 11 10 35 21  
## 6 Evangel… 575 869 1064 982 881 1486 949  
## 7 Hindu 1 9 7 9 11 34 47  
## 8 Histori… 228 244 236 238 197 223 131  
## 9 Jehovah… 20 27 24 24 21 30 15  
## 10 Jewish 19 19 25 25 30 95 69  
## 11 Mainlin… 289 495 619 655 651 1107 939  
## 12 Mormon 29 40 48 51 56 112 85  
## 13 Muslim 6 7 9 10 9 23 16  
## 14 Orthodox 13 17 23 32 32 47 38  
## 15 Other C… 9 7 11 13 13 14 18  
## 16 Other F… 20 33 40 46 49 63 46  
## 17 Other W… 5 2 3 4 2 7 3  
## 18 Unaffil… 217 299 374 365 341 528 407  
## # ℹ 3 more variables: `$100-150k` <dbl>, `>150k` <dbl>,  
## # `Don't know/refused` <dbl>

relig\_income%>%  
 pivot\_longer(-religion,   
 names\_to = "income range",  
 values\_to = "total")

## # A tibble: 180 × 3  
## religion `income range` total  
## <chr> <chr> <dbl>  
## 1 Agnostic <$10k 27  
## 2 Agnostic $10-20k 34  
## 3 Agnostic $20-30k 60  
## 4 Agnostic $30-40k 81  
## 5 Agnostic $40-50k 76  
## 6 Agnostic $50-75k 137  
## 7 Agnostic $75-100k 122  
## 8 Agnostic $100-150k 109  
## 9 Agnostic >150k 84  
## 10 Agnostic Don't know/refused 96  
## # ℹ 170 more rows

billboard <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab9/data/billboard.csv")

## Rows: 317 Columns: 79  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): artist, track  
## dbl (65): wk1, wk2, wk3, wk4, wk5, wk6, wk7, wk8, wk9, wk10, wk11, wk12, wk...  
## lgl (11): wk66, wk67, wk68, wk69, wk70, wk71, wk72, wk73, wk74, wk75, wk76  
## date (1): date.entered  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

billboard

## # A tibble: 317 × 79  
## artist track date.entered wk1 wk2 wk3 wk4 wk5 wk6 wk7 wk8  
## <chr> <chr> <date> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 2 Pac Baby… 2000-02-26 87 82 72 77 87 94 99 NA  
## 2 2Ge+her The … 2000-09-02 91 87 92 NA NA NA NA NA  
## 3 3 Doors D… Kryp… 2000-04-08 81 70 68 67 66 57 54 53  
## 4 3 Doors D… Loser 2000-10-21 76 76 72 69 67 65 55 59  
## 5 504 Boyz Wobb… 2000-04-15 57 34 25 17 17 31 36 49  
## 6 98^0 Give… 2000-08-19 51 39 34 26 26 19 2 2  
## 7 A\*Teens Danc… 2000-07-08 97 97 96 95 100 NA NA NA  
## 8 Aaliyah I Do… 2000-01-29 84 62 51 41 38 35 35 38  
## 9 Aaliyah Try … 2000-03-18 59 53 38 28 21 18 16 14  
## 10 Adams, Yo… Open… 2000-08-26 76 76 74 69 68 67 61 58  
## # ℹ 307 more rows  
## # ℹ 68 more variables: wk9 <dbl>, wk10 <dbl>, wk11 <dbl>, wk12 <dbl>,  
## # wk13 <dbl>, wk14 <dbl>, wk15 <dbl>, wk16 <dbl>, wk17 <dbl>, wk18 <dbl>,  
## # wk19 <dbl>, wk20 <dbl>, wk21 <dbl>, wk22 <dbl>, wk23 <dbl>, wk24 <dbl>,  
## # wk25 <dbl>, wk26 <dbl>, wk27 <dbl>, wk28 <dbl>, wk29 <dbl>, wk30 <dbl>,  
## # wk31 <dbl>, wk32 <dbl>, wk33 <dbl>, wk34 <dbl>, wk35 <dbl>, wk36 <dbl>,  
## # wk37 <dbl>, wk38 <dbl>, wk39 <dbl>, wk40 <dbl>, wk41 <dbl>, wk42 <dbl>, …

Some (but not all) of the column names are data. We also have NA’s. ## Solution 1: specify a range of columns that you want to pivot.

billboard2 <-   
 billboard %>%   
 pivot\_longer(wk1:wk76, # a range of columns that need to be tidy  
 names\_to = "week",  
 values\_to = "rank",   
 values\_drop\_na = TRUE #this will drop the NA's  
 )  
billboard2

## # A tibble: 5,307 × 5  
## artist track date.entered week rank  
## <chr> <chr> <date> <chr> <dbl>  
## 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk1 87  
## 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk2 82  
## 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk3 72  
## 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk4 77  
## 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk5 87  
## 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk6 94  
## 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk7 99  
## 8 2Ge+her The Hardest Part Of ... 2000-09-02 wk1 91  
## 9 2Ge+her The Hardest Part Of ... 2000-09-02 wk2 87  
## 10 2Ge+her The Hardest Part Of ... 2000-09-02 wk3 92  
## # ℹ 5,297 more rows

## Solution 2: OR, specify columns that you want to stay fixed.

billboard3 <-   
 billboard %>%   
 pivot\_longer(-c(artist, track, date.entered), #specific columns to stay fixed  
 names\_to = "week", # clean untidy  
 values\_to = "rank",  
 values\_drop\_na = TRUE  
 )  
billboard3

## # A tibble: 5,307 × 5  
## artist track date.entered week rank  
## <chr> <chr> <date> <chr> <dbl>  
## 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk1 87  
## 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk2 82  
## 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk3 72  
## 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk4 77  
## 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk5 87  
## 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk6 94  
## 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 wk7 99  
## 8 2Ge+her The Hardest Part Of ... 2000-09-02 wk1 91  
## 9 2Ge+her The Hardest Part Of ... 2000-09-02 wk2 87  
## 10 2Ge+her The Hardest Part Of ... 2000-09-02 wk3 92  
## # ℹ 5,297 more rows

## Solution 3: identify columns by a prefix, remove the prefix and all NA’s.

billboard4 <- billboard %>%   
 pivot\_longer(  
 cols = starts\_with("wk"), #columns that start with "wk" that untidy  
 names\_to = "week",  
 names\_prefix = "wk",  
 values\_to = "rank",  
 values\_drop\_na = TRUE); billboard4

## # A tibble: 5,307 × 5  
## artist track date.entered week rank  
## <chr> <chr> <date> <chr> <dbl>  
## 1 2 Pac Baby Don't Cry (Keep... 2000-02-26 1 87  
## 2 2 Pac Baby Don't Cry (Keep... 2000-02-26 2 82  
## 3 2 Pac Baby Don't Cry (Keep... 2000-02-26 3 72  
## 4 2 Pac Baby Don't Cry (Keep... 2000-02-26 4 77  
## 5 2 Pac Baby Don't Cry (Keep... 2000-02-26 5 87  
## 6 2 Pac Baby Don't Cry (Keep... 2000-02-26 6 94  
## 7 2 Pac Baby Don't Cry (Keep... 2000-02-26 7 99  
## 8 2Ge+her The Hardest Part Of ... 2000-09-02 1 91  
## 9 2Ge+her The Hardest Part Of ... 2000-09-02 2 87  
## 10 2Ge+her The Hardest Part Of ... 2000-09-02 3 92  
## # ℹ 5,297 more rows

## more than one variable in a cols — splite the col into multiple

qpcr\_untidy <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab9/data/qpcr\_untidy.csv");qpcr\_untidy

## Rows: 5 Columns: 10  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (1): gene  
## dbl (9): exp1\_rep1, exp1\_rep2, exp1\_rep3, exp2\_rep1, exp2\_rep2, exp2\_rep3, e...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

## # A tibble: 5 × 10  
## gene exp1\_rep1 exp1\_rep2 exp1\_rep3 exp2\_rep1 exp2\_rep2 exp2\_rep3 exp3\_rep1  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 A 21.7 19.8 20.7 18.3 20.4 17.6 20.6  
## 2 B 24.3 24.8 25.2 26.0 29.9 26.4 25.4  
## 3 C 20.7 21.5 21.3 25.5 18.7 22.3 21.9  
## 4 D 26.9 28.0 27.7 33.1 24.3 28.9 28.5  
## 5 E 25.0 22.7 23.8 21.1 23.4 20.2 23.7  
## # ℹ 2 more variables: exp3\_rep2 <dbl>, exp3\_rep3 <dbl>

#qpcr\_untidy %>%   
# pivot\_longer(  
# exp1\_rep1:exp3\_rep3,  
# names\_to= c("experiment", "replicate"),# name two new variables  
# names\_sep="\_", #pull apart 符号是链接两个部分的该符号。(separate variables name)  
# values\_to="mRNA\_expression"  
# )  
  
#heartrate3 <- heartrate2 %>%   
# separate(patient, into=c("patient", "sex"), sep="\_") %>% # separate obs names  
# pivot\_longer(-c(patient, sex),  
# names\_to = "drug",  
# values\_to = "heartrate")   
  
#heartrate3 %>%   
# unite(patient\_sex, "patient", "sex", sep=" ") # reunite the variables and obs ## unite(unitedvariables' name, var1, var2, sep = “链接obs的符号”)

# 9.2 Tidyr 2: pivot\_wider() You usepivot\_wider()when you have an observation scattered across multiple rows. ## Rules: +pivot\_wider(names\_from, values\_from) +names\_from- Values in thenames\_fromcolumn will become new column names +values\_from- Cell values will be taken from thevalues\_from` column

tb\_data <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab9/data/tb\_data.csv")

## Rows: 12 Columns: 4  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): country, key  
## dbl (2): year, value  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

tb\_data

## # A tibble: 12 × 4  
## country year key value  
## <chr> <dbl> <chr> <dbl>  
## 1 Afghanistan 1999 cases 745  
## 2 Afghanistan 1999 population 19987071  
## 3 Afghanistan 2000 cases 2666  
## 4 Afghanistan 2000 population 20595360  
## 5 Brazil 1999 cases 37737  
## 6 Brazil 1999 population 172006362  
## 7 Brazil 2000 cases 80488  
## 8 Brazil 2000 population 174504898  
## 9 China 1999 cases 212258  
## 10 China 1999 population 1272915272  
## 11 China 2000 cases 213766  
## 12 China 2000 population 1280428583

tb\_data%>%  
 pivot\_wider(names\_from = "key", values\_from = "value")

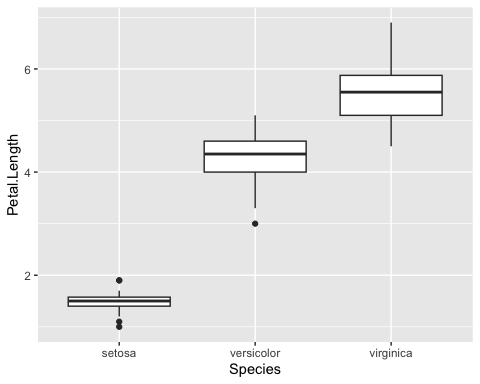
## # A tibble: 6 × 4  
## country year cases population  
## <chr> <dbl> <dbl> <dbl>  
## 1 Afghanistan 1999 745 19987071  
## 2 Afghanistan 2000 2666 20595360  
## 3 Brazil 1999 37737 172006362  
## 4 Brazil 2000 80488 174504898  
## 5 China 1999 212258 1272915272  
## 6 China 2000 213766 1280428583

# 10.1 ggplot - part1: plot= data + geom\_ + aesthetics.

1. Understand and apply the syntax of building plots using ggplot2.
2. Build a boxplot using ggplot2.
3. Build a scatterplot using ggplot2.
4. Build a barplot using ggplot2 and show the difference between stat=count and stat=identity.

### boxplot and scartter plot

library(tidyverse)  
library(naniar)  
library(janitor)  
  
### creating canvas for plot in first 2 lines  
ggplot(data=iris, #specify the data  
 mapping=aes(x=Species, y=Petal.Length))+ #map the aesthetics  
 geom\_boxplot() #add the plot type

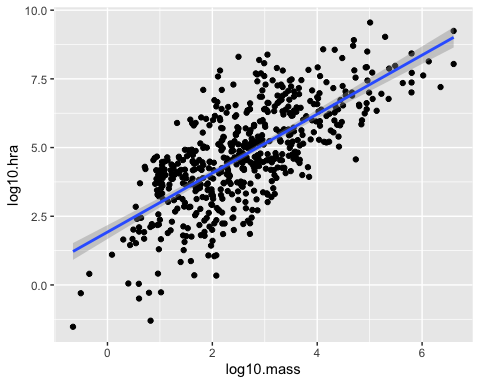


## scartter plot  
homerange <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab10/data/Tamburelloetal\_HomeRangeDatabase.csv")

## Rows: 569 Columns: 24  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (16): taxon, common.name, class, order, family, genus, species, primarym...  
## dbl (8): mean.mass.g, log10.mass, mean.hra.m2, log10.hra, dimension, preyma...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

ggplot(data=homerange, #specify the data  
 mapping=aes(x=log10.mass, y=log10.hra))+ #map the aesthetics  
 geom\_point(na.rm = T)+ #add the plot type  
 geom\_smooth(method=lm, se=T) #add a regression line

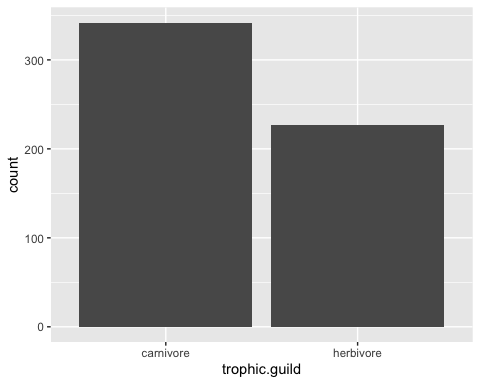
## `geom\_smooth()` using formula = 'y ~ x'



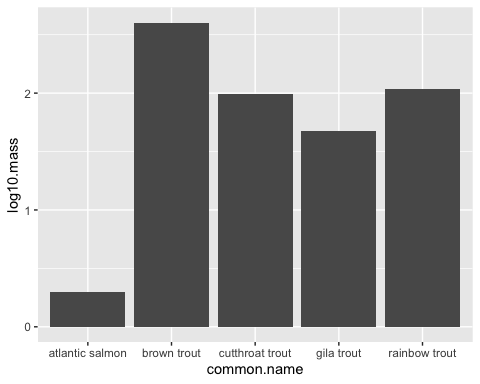
# method=lm, se=F, na.rm=F

### bar plot

homerange %>%   
 ggplot(aes(x=trophic.guild)) +   
 geom\_bar() #good for counts



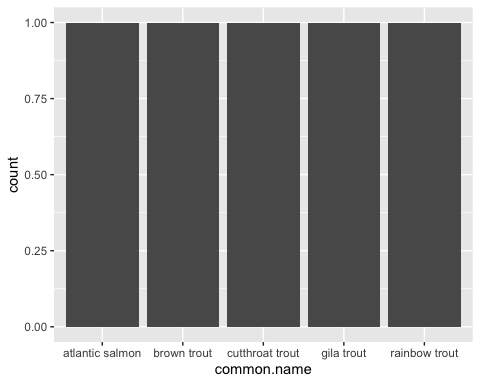
homerange %>%   
 filter(family=="salmonidae") %>%  
 select(common.name, log10.mass) %>%   
 ggplot(aes(x=common.name, y=log10.mass))+ #notice the switch in x and y  
 geom\_col()#+ # allow to specify the x and y axis



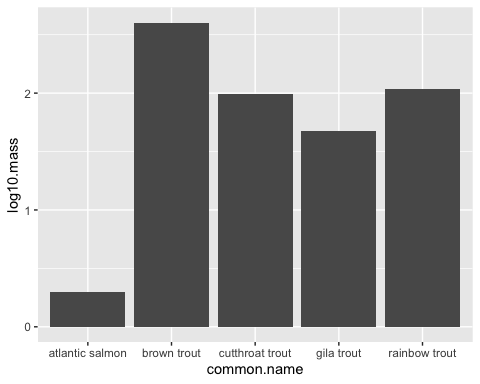
#coord\_flip() # change coordination

## geom\_bar() with stat=“identity” stat=“identity” allows us to map a variable to the y-axis so that we aren’t restricted to counts.

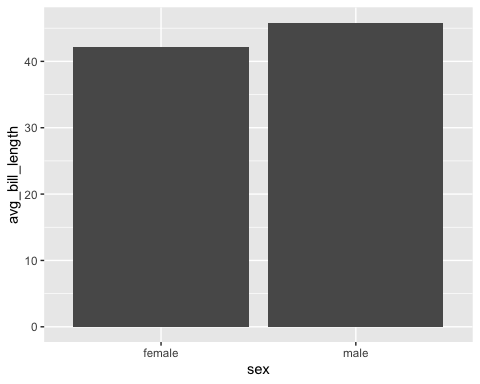
homerange %>%   
 filter(family=="salmonidae") %>%   
 ggplot(aes(x=common.name))+  
 geom\_bar()



homerange %>%   
 filter(family=="salmonidae") %>%   
 ggplot(aes(x=common.name, y=log10.mass))+   
 geom\_bar(stat="identity") # allows to counting y grouped by common name



library(palmerpenguins)  
penguins %>%   
 filter(sex != "NA")%>% # pull out NAs  
 group\_by(sex)%>%  
 summarize(avg\_bill\_length = mean(bill\_length\_mm, na.rm = T))%>%   
 ggplot(aes(x=sex, y = avg\_bill\_length))+  
 geom\_col()

 # 11.1 Data Visualization: ggplot part 2

options(scipen=999)#cancels scientific notation for the session  
scale\_y\_log10() #zoom in plot

## <ScaleContinuousPosition>  
## Range:   
## Limits: 0 -- 1

#life\_history%>%  
# group\_by(order)%>%  
# summarize(mean\_mass = mean(mass, na.rm = T))%>%  
# ggplot(aes(x = reorder(order, mean\_mass), y = mean\_mass))+ # allow us to reorder order by mean\_mass  
# geom\_col()+  
# coord\_flip()+  
# scale\_y\_log10()

### adjust position of title

#labs(title="Elephant Age vs. Height", # add title  
# x="Age(years)",   
# y="Height(cm")+  
# theme(plot.title = element\_text(size = rel(1.5), hjust = 0.5)) # remove the title to the middle and bigger, hjust is horizontal adjust

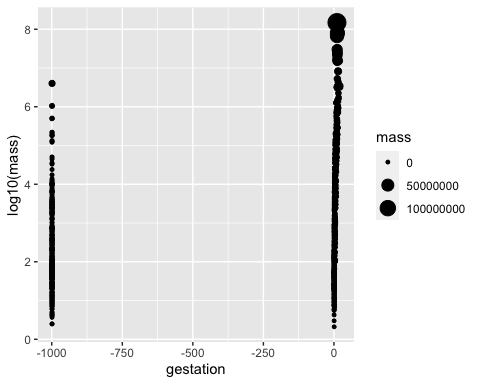
### fill:a common grouping option; show color based on different group in sex

#elephants %>%   
# ggplot(aes(x=sex, fill=sex))+  
# geom\_bar()

### size: adjusts the size of points relative to a continuous variable.

life\_history %>%   
 ggplot(aes(x=gestation, y=log10(mass), size=mass))+  
 geom\_point(na.rm=T)

## Warning in FUN(X[[i]], ...): NaNs produced  
  
## Warning in FUN(X[[i]], ...): NaNs produced



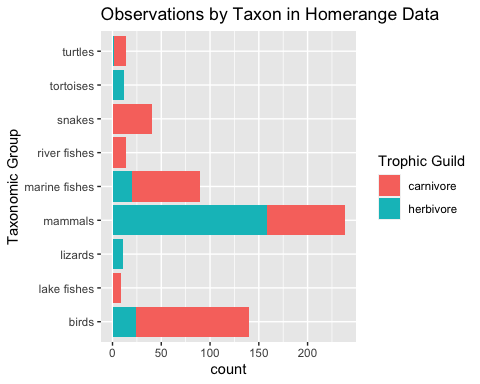
# geom\_point(size=1.25)  
#### adjust point size  
  
  
# geom\_point(aes(color=thermoregulation), size=1.5)   
#### color the points by a categorical variable.  
  
  
# geom\_point(aes(shape=thermoregulation, color=thermoregulation), size=1.5) ##### change shape& we use color here and not fill because we are not filling a shape

### counts & option2 also a stacked bar plot that shows the proportion

#option 1  
homerange %>% count(taxon, sort=T)

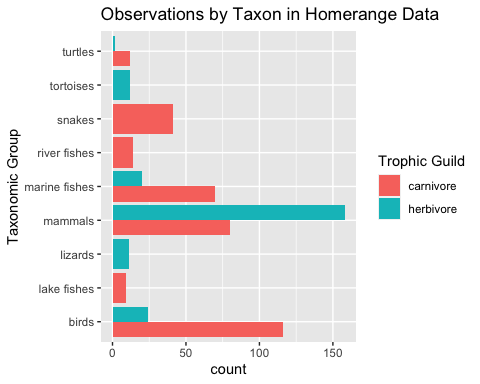
## # A tibble: 9 × 2  
## taxon n  
## <chr> <int>  
## 1 mammals 238  
## 2 birds 140  
## 3 marine fishes 90  
## 4 snakes 41  
## 5 river fishes 14  
## 6 turtles 14  
## 7 tortoises 12  
## 8 lizards 11  
## 9 lake fishes 9

# option 2  
homerange %>%   
 ggplot(aes(x = taxon, fill = trophic.guild)) +   
 geom\_bar() +  
 coord\_flip() +  
 labs(title = "Observations by Taxon in Homerange Data",  
 x = "Taxonomic Group",  
 fill = "Trophic Guild") # legend tittle



### geom\_bar(position = “dodge”) –>side-by-side bar plot for each taxon by specifying position=“dodge”

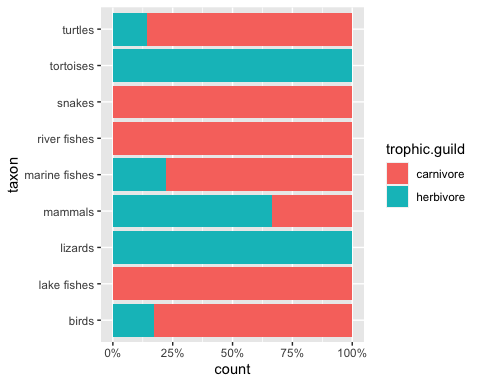
homerange %>%   
 ggplot(aes(x = taxon, fill = trophic.guild)) +   
 geom\_bar(position = "dodge") +  
 coord\_flip() +  
 labs(title = "Observations by Taxon in Homerange Data",  
 x = "Taxonomic Group",  
 fill = "Trophic Guild") # legend tittle



#theme(axis.text.x = element\_text(angle = 60, hjust = 1)) + #angle is the angle of the text if taxon names are crowded for vertical plot

### percentage bar plot

homerange %>%   
 ggplot(aes(x = taxon, fill = trophic.guild))+  
 geom\_bar(position = position\_fill())+   
 scale\_y\_continuous(labels = scales::percent)+  
 coord\_flip()



# 12.1 Data Visualization: ggplot part 3

#library(tidyverse)  
#library(RColorBrewer)  
#library(paletteer)  
#library(janitor)  
#deserts <- read\_csv("/Users/zhuoyawang/desktop/github/BIS15W2024\_zywang/lab10/data/surveys\_complete.csv")

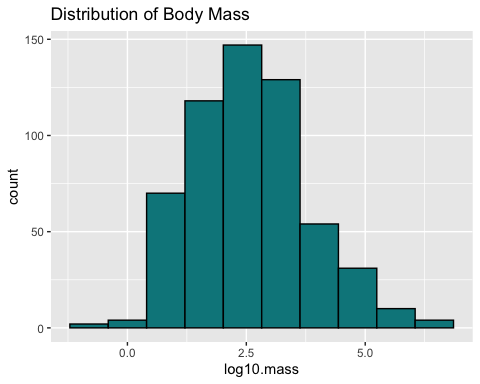
## line plot: it is need to make year as.factor(year)

#deserts%>%  
# filter(species\_id == "DM" | species\_id == "DS")%>%  
# mutate(year = as.factor(year))%>%  
# group\_by(year, species\_id)%>%  
# summarize(n = n(), .groups= "keep")%>% # make it more clean  
# pivot\_wider(names\_from = species\_id, values\_from = n)

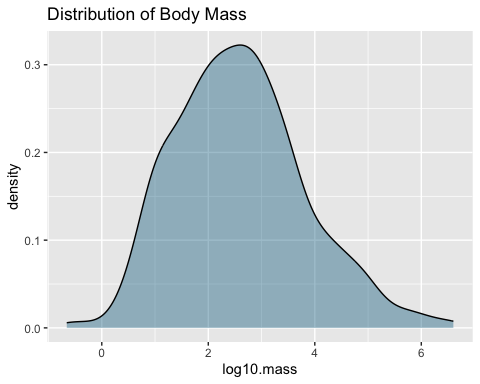
#gapminder%>%  
# group\_by(year, continent)%>%  
# summarize(lc = mean(lifeExp, na.rm = T),.groups= "keep")%>%  
# ggplot(aes(x = year, y = lc, color = continent))+  
# geom\_line()+  
# labs(title = "Life Expectancy by Continent",  
# x = "Year",  
# y = "Average Life Expectancy")

### histogram; color ->boxline, fill ->inside box, bins-> box size

homerange %>%   
 ggplot(aes(x = log10.mass)) +  
 geom\_histogram(color = "black", fill = "turquoise4", bins=10)+  
 labs(title = "Distribution of Body Mass")

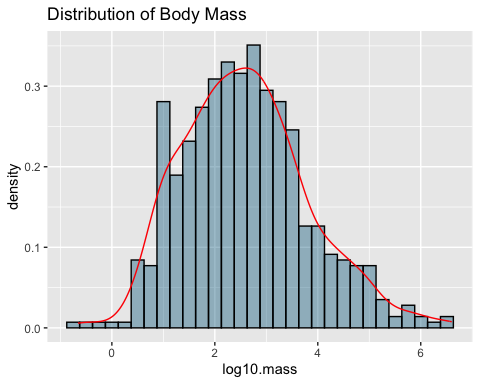
 ### Density plot: geom\_density; alpha -> transparency of plot.

homerange %>%   
 ggplot(aes(x = log10.mass)) +  
 geom\_density(fill="deepskyblue4", alpha =0.4, color = "black")+ #alpha is the transparency  
 labs(title = "Distribution of Body Mass")



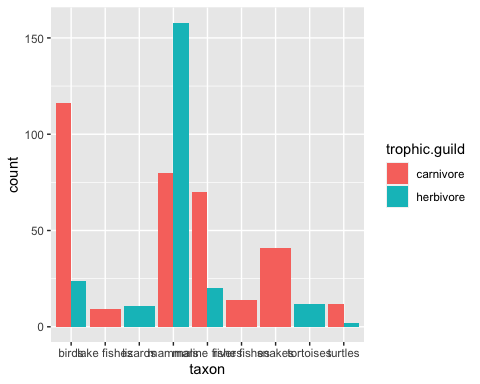
homerange %>%   
 ggplot(aes(x=log10.mass)) +  
 geom\_histogram(aes(y = after\_stat(density)), fill = "deepskyblue4", alpha = 0.4, color = "black")+  
 geom\_density(color = "red")+ # density line  
 labs(title = "Distribution of Body Mass")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



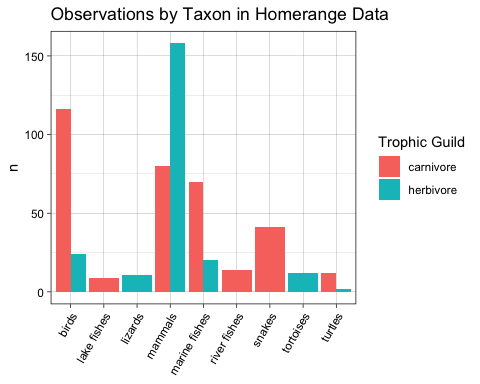
# 12.2

p <- homerange %>%   
 ggplot(aes(x=taxon, fill=trophic.guild))+  
 geom\_bar(na.rm=T, position="dodge");p

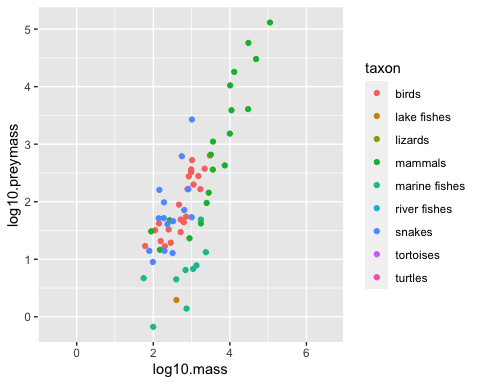


### linedraw : adding another layer

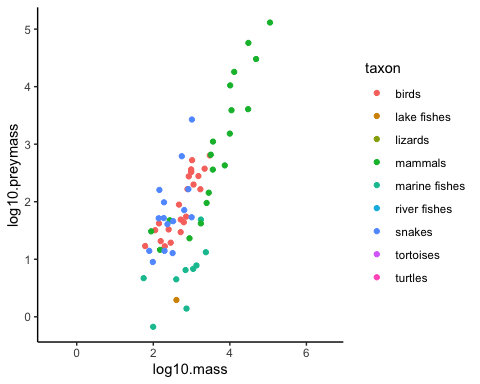
p + theme\_linedraw()+  
 theme(axis.text.x = element\_text(angle = 60, hjust=1))+  
 labs(title = "Observations by Taxon in Homerange Data",  
 x = NULL,  
 y= "n",  
 fill= "Trophic Guild")



q <- homerange %>%   
 ggplot(aes(x=log10.mass, y=log10.preymass, color=taxon))+  
 geom\_point(na.rm=T);q

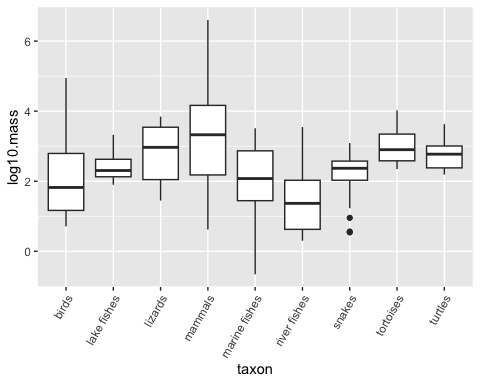


q + theme\_classic()



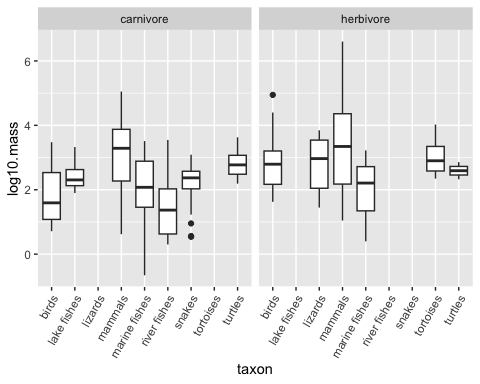
## Faceting

homerange %>%   
 ggplot(aes(x=taxon, y=log10.mass))+  
 geom\_boxplot()+  
 theme(axis.text.x = element\_text(angle = 60, hjust=1))



## facet\_wrap makes a ribbon of panels by a specified categorical variable

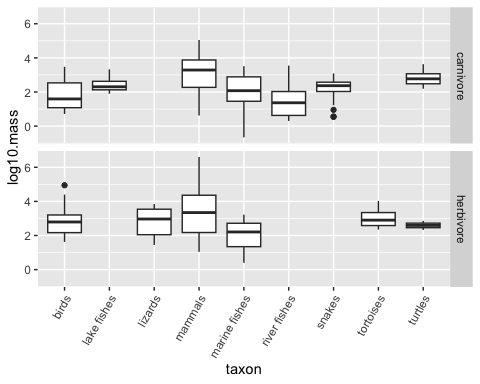
homerange %>%   
 ggplot(aes(x=taxon, y=log10.mass))+  
 geom\_boxplot()+  
 facet\_wrap(~trophic.guild, ncol=2)+  
 theme(axis.text.x = element\_text(angle = 60, hjust=1))



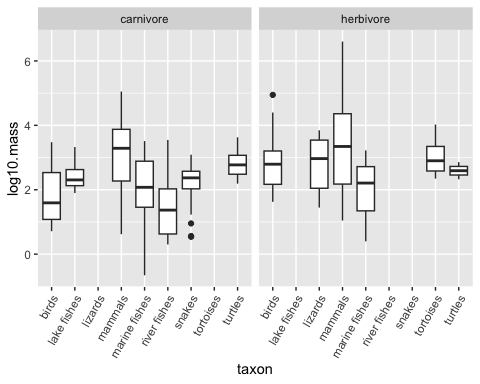
## facet\_grid: control over the faceted variable; it can be arranged in rows or columns.

## rows~columns.

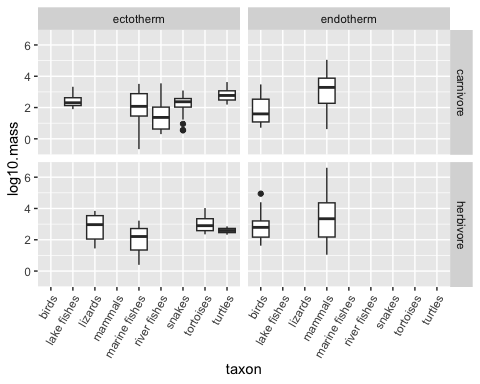
homerange %>%   
 ggplot(aes(x=taxon, y=log10.mass))+  
 geom\_boxplot()+  
 facet\_grid(trophic.guild~.)+ # in rows  
 theme(axis.text.x = element\_text(angle = 60, hjust=1))



homerange %>%   
 ggplot(aes(x=taxon, y=log10.mass))+  
 geom\_boxplot()+  
 facet\_grid(.~trophic.guild)+ # in col   
 theme(axis.text.x = element\_text(angle = 60, hjust=1))



homerange %>%   
 ggplot(aes(x=taxon, y=log10.mass))+  
 geom\_boxplot()+  
 facet\_grid(trophic.guild~thermoregulation)+  
 theme(axis.text.x = element\_text(angle = 60, hjust=1))



## HW11 review

#top5 <- gapminder%>%  
# filter(year %in% c(1952, 2007))%>%  
# group\_by(country)%>%  
# summarize(start = first(pop),  
# end = last(pop),  
# pop\_growth = (end-start))%>%  
# arrange(desc(pop\_growth))%>%  
# top\_n(5,pop\_growth);top5

#gapminder%>%  
# filter(year %in%c(1952, 2007) & country %in% top5$country)%>%  
# ggplot(aes(x = reorder(country, pop), y = pop, fill = year))+  
# geom\_bar(stat = "identity", position = "dodge")+  
# coord\_flip()+  
# labs(title = "Top 5 Population Growth Countries", x = "population", y = "country")