tidying

Saneesh

1/24/2022

packages

```
# install.packages ('gapminder')
library(gapminder)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

shortcuts

alt+- will add <-
shift+ctrl+c to add # infront of a line
'—-' for a header, so it is easy to navigate through the script
command +shift + m
ctrl+alt+i for new code chunk # syntax Plain text
End a line with two spaces to start a new paragraph.
italics and *italics*
bold and **bold**
superscript^2
~~strikethrough~~
link to rstudio

logical operations

1==1 # equality
1!=3 #unequal
13<14 #13 smaller than 14
14>13 #14 bigger than 13
12>=0 #12 greater or equal to zero
12<=3 #12 smaller or equal to zero

creating data.frame

family

```
name <- c('saneesh', 'sanusha', 'appu', 'kishan')
weight <- c(63,48, 20, NA)
height <- c(164, 150, NA, 75)
family <- data.frame(name, weight, height)
family %>% as_tibble()
```

```
## # A tibble: 4 x 3
##   name    weight height
##   <chr>    <dbl>  <dbl>
## 1 saneesh     63    164
## 2 sanusha     48    150
## 3 appu        20     NA
## 4 kishan      NA     75
```

```
library(tidyverse)
data <- data.frame(sex=c(rep('female', 10), rep('male', 8)),
                   score=c(rnorm(n= 10, mean = 7.56, sd = 1.978), rnorm(n= 8, mean=7.75, sd= 1.631)))

data
```

data frame with unequal values 10 and 8

```
##       sex     score
## 1  female  4.864086
## 2  female  8.535747
## 3  female  8.029077
## 4  female  7.258620
## 5  female  9.315173
## 6  female  5.233860
## 7  female  9.459136
## 8  female 12.282502
## 9  female  7.463015
## 10 female  5.839924
## 11   male  6.379641
## 12   male  7.715882
## 13   male  8.117082
## 14   male  7.206449
## 15   male  8.282078
## 16   male  9.186558
## 17   male  9.292400
## 18   male  9.139804
```

```
data %>% group_by(sex) %>%
  summarise(score= n()) %>%
  mutate(freq=score/sum(score)*100)
```

```
## # A tibble: 2 x 3
##   sex    score  freq
##   <chr>  <int> <dbl>
## 1 female    10  55.6
## 2 male       8  44.4
```

is.na

```
# identify location of NAs in vector
which(is.na(family))
```

```
## [1]  8 11
```

```
colSums(is.na(family))
```

```
##   name weight height
##      0      1      1
```

replace na

```
mat <- matrix(sample(c(NA, 1:5), 50, replace = TRUE), 5)
df <- as.data.frame(mat)
df %>% replace(is.na(.), 0)%>% view()
```

clean names

```
# install.packages('janitor')
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

```
id <- (c(1,1,2,2,3,3))
Country <- c('Angola', 'Angola','Botswana', 'Botswana','Zimbabwe','Zimbabwe')
year <- c('2006', '2007', '2008', '2009', '2010', '2006')
bank.ratio <- c(24,25,38,34,42,49)
Reserve.ratio <- c(77,59,64,65,57,86)
broad.money <- c(163,188,317,361,150,288)
```

```
bank <- data.frame(id, Country, year, bank.ratio, Reserve.ratio,broad.money)

bank %>% view()
  as_tibble()
```

```
## Warning: The `x` argument of `as_tibble()` can't be missing as of tibble 3.0.0.

## # A tibble: 0 x 0
```

```
bank <- bank %>% clean_names() # replaced . with _

glimpse(bank)
```

```
## Rows: 6
## Columns: 6
## $ id            <dbl> 1, 1, 2, 2, 3, 3
## $ country       <chr> "Angola", "Angola", "Botswana", "Botswana", "Zimbabwe", ~
## $ year          <chr> "2006", "2007", "2008", "2009", "2010", "2006"
## $ bank_ratio    <dbl> 24, 25, 38, 34, 42, 49
## $ reserve_ratio <dbl> 77, 59, 64, 65, 57, 86
## $ broad_money   <dbl> 163, 188, 317, 361, 150, 288
```

```
bank <- bank %>% clean_names() # replaced . with _
```

filter bank data frame below such that it retains a country if a given id is satisfied e.g. filtering a data frame that has countries with id 1 and 2 only

```
bank %>%
  filter(id%in% c(1,2)) %>%
  as_tibble()
```

```
## # A tibble: 4 x 6
##      id country  year  bank_ratio reserve_ratio broad_money
##   <dbl> <chr>    <chr>      <dbl>         <dbl>       <dbl>
## 1     1 Angola   2006          24            77         163
## 2     1 Angola   2007          25            59         188
## 3     2 Botswana 2008          38            64         317
## 4     2 Botswana 2009          34            65         361
```

summarise fund available with each countries

```
bank %>%
  group_by(country) %>%
  summarise(fund=sum(broad_money)) %>%
  as_tibble()
```

```
## # A tibble: 3 x 2
##   country   fund
##   <chr>    <dbl>
## 1 Angola     351
## 2 Botswana   678
## 3 Zimbabwe   438
```

count/ frequency

```
mtcars %>%
  count(am) %>%
  as_tibble()
```

```
## # A tibble: 2 x 2
##      am     n
##   <dbl> <int>
## 1     0    19
## 2     1    13
```

```
mtcars %>%
  count(gear)
```

```
##   gear  n
## 1    3 15
## 2    4 12
## 3    5  5
```

rename column

column: new name= old name

```
iris %>%
  rename(sep.len=Sepal.Length)
```

```
##     sep.len Sepal.Width Petal.Length Petal.Width    Species
## 1       5.1         3.5          1.4         0.2     setosa
## 2       4.9         3.0          1.4         0.2     setosa
## 3       4.7         3.2          1.3         0.2     setosa
## 4       4.6         3.1          1.5         0.2     setosa
## 5       5.0         3.6          1.4         0.2     setosa
## 6       5.4         3.9          1.7         0.4     setosa
## 7       4.6         3.4          1.4         0.3     setosa
## 8       5.0         3.4          1.5         0.2     setosa
## 9       4.4         2.9          1.4         0.2     setosa
## 10      4.9         3.1          1.5         0.1     setosa
## 11      5.4         3.7          1.5         0.2     setosa
## 12      4.8         3.4          1.6         0.2     setosa
## 13      4.8         3.0          1.4         0.1     setosa
## 14      4.3         3.0          1.1         0.1     setosa
## 15      5.8         4.0          1.2         0.2     setosa
## 16      5.7         4.4          1.5         0.4     setosa
## 17      5.4         3.9          1.3         0.4     setosa
## 18      5.1         3.5          1.4         0.3     setosa
## 19      5.7         3.8          1.7         0.3     setosa
## 20      5.1         3.8          1.5         0.3     setosa
## 21      5.4         3.4          1.7         0.2     setosa
```

```
## 22      5.1         3.7          1.5         0.4     setosa
## 23      4.6         3.6          1.0         0.2     setosa
## 24      5.1         3.3          1.7         0.5     setosa
## 25      4.8         3.4          1.9         0.2     setosa
## 26      5.0         3.0          1.6         0.2     setosa
## 27      5.0         3.4          1.6         0.4     setosa
## 28      5.2         3.5          1.5         0.2     setosa
## 29      5.2         3.4          1.4         0.2     setosa
## 30      4.7         3.2          1.6         0.2     setosa
## 31      4.8         3.1          1.6         0.2     setosa
## 32      5.4         3.4          1.5         0.4     setosa
## 33      5.2         4.1          1.5         0.1     setosa
## 34      5.5         4.2          1.4         0.2     setosa
## 35      4.9         3.1          1.5         0.2     setosa
## 36      5.0         3.2          1.2         0.2     setosa
## 37      5.5         3.5          1.3         0.2     setosa
## 38      4.9         3.6          1.4         0.1     setosa
## 39      4.4         3.0          1.3         0.2     setosa
## 40      5.1         3.4          1.5         0.2     setosa
## 41      5.0         3.5          1.3         0.3     setosa
## 42      4.5         2.3          1.3         0.3     setosa
## 43      4.4         3.2          1.3         0.2     setosa
## 44      5.0         3.5          1.6         0.6     setosa
## 45      5.1         3.8          1.9         0.4     setosa
## 46      4.8         3.0          1.4         0.3     setosa
## 47      5.1         3.8          1.6         0.2     setosa
## 48      4.6         3.2          1.4         0.2     setosa
## 49      5.3         3.7          1.5         0.2     setosa
## 50      5.0         3.3          1.4         0.2     setosa
## 51      7.0         3.2          4.7         1.4 versicolor
## 52      6.4         3.2          4.5         1.5 versicolor
## 53      6.9         3.1          4.9         1.5 versicolor
## 54      5.5         2.3          4.0         1.3 versicolor
## 55      6.5         2.8          4.6         1.5 versicolor
## 56      5.7         2.8          4.5         1.3 versicolor
## 57      6.3         3.3          4.7         1.6 versicolor
## 58      4.9         2.4          3.3         1.0 versicolor
## 59      6.6         2.9          4.6         1.3 versicolor
## 60      5.2         2.7          3.9         1.4 versicolor
## 61      5.0         2.0          3.5         1.0 versicolor
## 62      5.9         3.0          4.2         1.5 versicolor
## 63      6.0         2.2          4.0         1.0 versicolor
## 64      6.1         2.9          4.7         1.4 versicolor
## 65      5.6         2.9          3.6         1.3 versicolor
## 66      6.7         3.1          4.4         1.4 versicolor
## 67      5.6         3.0          4.5         1.5 versicolor
## 68      5.8         2.7          4.1         1.0 versicolor
## 69      6.2         2.2          4.5         1.5 versicolor
## 70      5.6         2.5          3.9         1.1 versicolor
## 71      5.9         3.2          4.8         1.8 versicolor
## 72      6.1         2.8          4.0         1.3 versicolor
## 73      6.3         2.5          4.9         1.5 versicolor
## 74      6.1         2.8          4.7         1.2 versicolor
## 75      6.4         2.9          4.3         1.3 versicolor
```

```
## 76      6.6         3.0          4.4         1.4 versicolor
## 77      6.8         2.8          4.8         1.4 versicolor
## 78      6.7         3.0          5.0         1.7 versicolor
## 79      6.0         2.9          4.5         1.5 versicolor
## 80      5.7         2.6          3.5         1.0 versicolor
## 81      5.5         2.4          3.8         1.1 versicolor
## 82      5.5         2.4          3.7         1.0 versicolor
## 83      5.8         2.7          3.9         1.2 versicolor
## 84      6.0         2.7          5.1         1.6 versicolor
## 85      5.4         3.0          4.5         1.5 versicolor
## 86      6.0         3.4          4.5         1.6 versicolor
## 87      6.7         3.1          4.7         1.5 versicolor
## 88      6.3         2.3          4.4         1.3 versicolor
## 89      5.6         3.0          4.1         1.3 versicolor
## 90      5.5         2.5          4.0         1.3 versicolor
## 91      5.5         2.6          4.4         1.2 versicolor
## 92      6.1         3.0          4.6         1.4 versicolor
## 93      5.8         2.6          4.0         1.2 versicolor
## 94      5.0         2.3          3.3         1.0 versicolor
## 95      5.6         2.7          4.2         1.3 versicolor
## 96      5.7         3.0          4.2         1.2 versicolor
## 97      5.7         2.9          4.2         1.3 versicolor
## 98      6.2         2.9          4.3         1.3 versicolor
## 99      5.1         2.5          3.0         1.1 versicolor
## 100     5.7         2.8          4.1         1.3 versicolor
## 101     6.3         3.3          6.0         2.5  virginica
## 102     5.8         2.7          5.1         1.9  virginica
## 103     7.1         3.0          5.9         2.1  virginica
## 104     6.3         2.9          5.6         1.8  virginica
## 105     6.5         3.0          5.8         2.2  virginica
## 106     7.6         3.0          6.6         2.1  virginica
## 107     4.9         2.5          4.5         1.7  virginica
## 108     7.3         2.9          6.3         1.8  virginica
## 109     6.7         2.5          5.8         1.8  virginica
## 110     7.2         3.6          6.1         2.5  virginica
## 111     6.5         3.2          5.1         2.0  virginica
## 112     6.4         2.7          5.3         1.9  virginica
## 113     6.8         3.0          5.5         2.1  virginica
## 114     5.7         2.5          5.0         2.0  virginica
## 115     5.8         2.8          5.1         2.4  virginica
## 116     6.4         3.2          5.3         2.3  virginica
## 117     6.5         3.0          5.5         1.8  virginica
## 118     7.7         3.8          6.7         2.2  virginica
## 119     7.7         2.6          6.9         2.3  virginica
## 120     6.0         2.2          5.0         1.5  virginica
## 121     6.9         3.2          5.7         2.3  virginica
## 122     5.6         2.8          4.9         2.0  virginica
## 123     7.7         2.8          6.7         2.0  virginica
## 124     6.3         2.7          4.9         1.8  virginica
## 125     6.7         3.3          5.7         2.1  virginica
## 126     7.2         3.2          6.0         1.8  virginica
## 127     6.2         2.8          4.8         1.8  virginica
## 128     6.1         3.0          4.9         1.8  virginica
## 129     6.4         2.8          5.6         2.1  virginica
```

```
## 130     7.2         3.0          5.8         1.6  virginica
## 131     7.4         2.8          6.1         1.9  virginica
## 132     7.9         3.8          6.4         2.0  virginica
## 133     6.4         2.8          5.6         2.2  virginica
## 134     6.3         2.8          5.1         1.5  virginica
## 135     6.1         2.6          5.6         1.4  virginica
## 136     7.7         3.0          6.1         2.3  virginica
## 137     6.3         3.4          5.6         2.4  virginica
## 138     6.4         3.1          5.5         1.8  virginica
## 139     6.0         3.0          4.8         1.8  virginica
## 140     6.9         3.1          5.4         2.1  virginica
## 141     6.7         3.1          5.6         2.4  virginica
## 142     6.9         3.1          5.1         2.3  virginica
## 143     5.8         2.7          5.1         1.9  virginica
## 144     6.8         3.2          5.9         2.3  virginica
## 145     6.7         3.3          5.7         2.5  virginica
## 146     6.7         3.0          5.2         2.3  virginica
## 147     6.3         2.5          5.0         1.9  virginica
## 148     6.5         3.0          5.2         2.0  virginica
## 149     6.2         3.4          5.4         2.3  virginica
## 150     5.9         3.0          5.1         1.8  virginica
```

gapminder

```
iris %>% as_tibble()
```

```
## # A tibble: 150 x 5
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
##           <dbl>       <dbl>        <dbl>       <dbl> <fct>
##  1          5.1         3.5          1.4         0.2 setosa
##  2          4.9         3            1.4         0.2 setosa
##  3          4.7         3.2          1.3         0.2 setosa
##  4          4.6         3.1          1.5         0.2 setosa
##  5          5           3.6          1.4         0.2 setosa
##  6          5.4         3.9          1.7         0.4 setosa
##  7          4.6         3.4          1.4         0.3 setosa
##  8          5           3.4          1.5         0.2 setosa
##  9          4.4         2.9          1.4         0.2 setosa
## 10          4.9         3.1          1.5         0.1 setosa
## # ... with 140 more rows
```

```
summary(gapminder)
```

```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
```

```
##  (Other)    :1632
##       pop              gdpPercap
##  Min.   :6.001e+04   Min.   :   241.2
##  1st Qu.:2.794e+06   1st Qu.:  1202.1
##  Median :7.024e+06   Median :  3531.8
##  Mean   :2.960e+07   Mean   :  7215.3
##  3rd Qu.:1.959e+07   3rd Qu.:  9325.5
##  Max.   :1.319e+09   Max.   :113523.1
##
```

```
str(gapminder)
```

```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
##  $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
##  $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

recode observation

change name of observation— mutate (variable=recode (variable, 'old name'='new name')))

```
gapminder %>%
 mutate(country=recode(country, 'India'='IND' )) %>%
  filter(country=='IND')
```

```
## # A tibble: 12 x 6
##    country continent  year lifeExp        pop gdpPercap
##    <fct>   <fct>     <int>   <dbl>      <int>     <dbl>
##  1 IND     Asia       1952    37.4  372000000      547.
##  2 IND     Asia       1957    40.2  409000000      590.
##  3 IND     Asia       1962    43.6  454000000      658.
##  4 IND     Asia       1967    47.2  506000000      701.
##  5 IND     Asia       1972    50.7  567000000      724.
##  6 IND     Asia       1977    54.2  634000000      813.
##  7 IND     Asia       1982    56.6  708000000      856.
##  8 IND     Asia       1987    58.6  788000000      977.
##  9 IND     Asia       1992    60.2  872000000     1164.
## 10 IND     Asia       1997    61.8  959000000     1459.
## 11 IND     Asia       2002    62.9 1034172547     1747.
## 12 IND     Asia       2007    64.7 1110396331     2452.
```

select

```
gapminder %>%
  select(year, country, gdpPercap)
```

```
## # A tibble: 1,704 x 3
##     year country     gdpPercap
##    <int> <fct>           <dbl>
##  1  1952 Afghanistan      779.
##  2  1957 Afghanistan      821.
##  3  1962 Afghanistan      853.
##  4  1967 Afghanistan      836.
##  5  1972 Afghanistan      740.
##  6  1977 Afghanistan      786.
##  7  1982 Afghanistan      978.
##  8  1987 Afghanistan      852.
##  9  1992 Afghanistan      649.
## 10  1997 Afghanistan      635.
## # ... with 1,694 more rows
```

```
msleep %>% select(starts_with("sleep"))
```

```
## # A tibble: 83 x 3
##    sleep_total sleep_rem sleep_cycle
##          <dbl>     <dbl>       <dbl>
##  1        12.1      NA         NA
##  2        17         1.8       NA
##  3        14.4       2.4       NA
##  4        14.9       2.3        0.133
##  5         4         0.7        0.667
##  6        14.4       2.2        0.767
##  7         8.7       1.4        0.383
##  8         7        NA         NA
##  9        10.1       2.9        0.333
## 10         3        NA         NA
## # ... with 73 more rows
```

filter

```
gapminder %>%
  select(year, country, lifeExp) %>%
  filter(country=="Eritrea", year>1950)
```

```
## # A tibble: 12 x 3
##     year country lifeExp
##    <int> <fct>     <dbl>
##  1  1952 Eritrea    35.9
##  2  1957 Eritrea    38.0
##  3  1962 Eritrea    40.2
##  4  1967 Eritrea    42.2
##  5  1972 Eritrea    44.1
##  6  1977 Eritrea    44.5
##  7  1982 Eritrea    43.9
##  8  1987 Eritrea    46.5
##  9  1992 Eritrea    50.0
```

```
## 10  1997 Eritrea    53.4
## 11  2002 Eritrea    55.2
## 12  2007 Eritrea    58.0
```

```
gapminder %>% filter(country=="Canada") %>% head(3) # from gapminder data filter country Canada and sho
```

```
## # A tibble: 3 x 6
##   country continent  year lifeExp      pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Canada  Americas   1952    68.8 14785584    11367.
## 2 Canada  Americas   1957    70.0 17010154    12490.
## 3 Canada  Americas   1962    71.3 18985849    13462.
```

```
gapminder %>% filter(country!="Oman") %>% head(3) # from gapminder data filter all the other countries
```

```
## # A tibble: 3 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
```

multiple conditions

```
gapminder %>%
  filter(country=="Oman" &
           year>1980 &
           year<=2000) %>% head(4)
```

```
## # A tibble: 4 x 6
##   country continent  year lifeExp     pop gdpPercap
##   <fct>   <fct>     <int>   <dbl>   <int>     <dbl>
## 1 Oman    Asia       1982    62.7 1301048    12955.
## 2 Oman    Asia       1987    67.7 1593882    18115.
## 3 Oman    Asia       1992    71.2 1915208    18617.
## 4 Oman    Asia       1997    72.5 2283635    19702.
```

```
gapminder %>%
  select(country, year) %>%
  filter(year>=1980, country=="India"|
           country=="Oman"|
           country=="Canada")
```

```
## # A tibble: 18 x 2
##    country  year
##    <fct>   <int>
##  1 Canada   1982
##  2 Canada   1987
##  3 Canada   1992
##  4 Canada   1997
##  5 Canada   2002
```

```
##  6 Canada   2007
##  7 India    1982
##  8 India    1987
##  9 India    1992
## 10 India    1997
## 11 India    2002
## 12 India    2007
## 13 Oman     1982
## 14 Oman     1987
## 15 Oman     1992
## 16 Oman     1997
## 17 Oman     2002
## 18 Oman     2007
```

```
gapminder %>% filter(country!="Oman") %>% head(3) # from gapminder data filter all the other countires
```

```
## # A tibble: 3 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
```

filter multipe using %in%

```
gapminder %>% filter(country %in% c('Hungary','Iceland', 'Mongolia')) %>% view()

target <- c('Hungary','Iceland', 'Mongolia')
gapminder %>% filter(country %in% target) %>% view()
```

```
friends <- data.frame(Names=c('Saneesh', 'Appu', 'Shruti', 'Aradhana', 'Arathi', 'James Bond'),
                      age=c(40,9, 25, 25, 25, 50))
# data frame is friends
# columns in friends are Names, Age, Height, etc.
# Colum Name have 'Saneesh', 'Appu', 'Shruti', 'Aradhana', 'Arathi', 'James Bond'
# We want to filter information related to Sanees and James Bond only, so we created a vector with
# these names in it.

target <- c('Appu', 'James Bond') #and then

friends %>% filter(Names %in% target)
```

```
##        Names age
## 1       Appu   9
## 2 James Bond  50
```

```
# or
friends %>% filter(Names== 'Appu'| Names== 'James Bond')
```

```
##        Names age
## 1       Appu   9
## 2 James Bond  50
```

```
# or
friends %>% filter(Names %in% c('Appu', 'James Bond'))
```

```
##        Names age
## 1       Appu   9
## 2 James Bond  50
```

drop

```
gapminder %>%
  select(-year,-pop) %>%
  head(5)
```

```
## # A tibble: 5 x 4
##   country     continent lifeExp gdpPercap
##   <fct>       <fct>       <dbl>     <dbl>
## 1 Afghanistan Asia         28.8      779.
## 2 Afghanistan Asia         30.3      821.
## 3 Afghanistan Asia         32.0      853.
## 4 Afghanistan Asia         34.0      836.
## 5 Afghanistan Asia         36.1      740.
```

group by & summarise

```
gapminder %>%
  filter(year==2007) %>%
  group_by(country) %>%
  summarise(meanLE=mean(lifeExp)) %>%
  arrange(meanLE,decreasing = TRUE)
```

```
## # A tibble: 142 x 2
##    country                  meanLE
##    <fct>                     <dbl>
##  1 Swaziland                  39.6
##  2 Mozambique                 42.1
##  3 Zambia                     42.4
##  4 Sierra Leone               42.6
##  5 Lesotho                    42.6
##  6 Angola                     42.7
##  7 Zimbabwe                   43.5
##  8 Afghanistan                43.8
##  9 Central African Republic   44.7
## 10 Liberia                    45.7
## # ... with 132 more rows
```

```
gapminder %>%
  group_by(country) %>%
  summarise(minLE=min(lifeExp)) %>%
  arrange(minLE,decreasing=FALSE)
```

```
## # A tibble: 142 x 2
##    country       minLE
##    <fct>         <dbl>
##  1 Rwanda         23.6
##  2 Afghanistan    28.8
##  3 Gambia         30
##  4 Angola         30.0
##  5 Sierra Leone   30.3
##  6 Cambodia       31.2
##  7 Mozambique     31.3
##  8 Burkina Faso   32.0
##  9 Guinea-Bissau  32.5
## 10 Yemen, Rep.    32.5
## # ... with 132 more rows
```

grouped by continent, then summarise two things, first n=n() number of rows in which each continent are or the size of each group, then the mean of the mean of the lifeExp variable.

```
gapminder %>%
  group_by(continent) %>%
  summarise(n=n(),
            meanLife=mean(lifeExp))
```

```
## # A tibble: 5 x 3
##   continent     n meanLife
##   <fct>     <int>    <dbl>
## 1 Africa      624     48.9
## 2 Americas    300     64.7
## 3 Asia        396     60.1
## 4 Europe      360     71.9
## 5 Oceania      24     74.3
```

```
gapminder %>%
  group_by(continent) %>%
  summarise(PopConti=sum(pop))
```

```
## # A tibble: 5 x 2
##   continent    PopConti
##   <fct>           <dbl>
## 1 Africa     6187585961
## 2 Americas   7351438499
## 3 Asia      30507333901
## 4 Europe     6181115304
## 5 Oceania     212992136
```

```
pets <- data.frame(names=c(rep('saneesh', 3), rep('appu', 2), 'sanusha'),
                   pet=c(rep('dog', 3), rep('cat', 2), 'tiger'), number=c(2,2,5,7,8,1),
                   size=c(rep('medium', 2), rep('small', 3), 'big'))

pets
```

```
##     names   pet number   size
## 1 saneesh   dog      2 medium
## 2 saneesh   dog      2 medium
## 3 saneesh   dog      5  small
## 4    appu   cat      7  small
## 5    appu   cat      8  small
## 6 sanusha tiger      1    big
```

```
library(tidyverse)

pets %>% group_by(pet, size) %>%
  summarise(totalpet= sum(number))
```

```
## `summarise()` has grouped output by 'pet'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 4 x 3
## # Groups:   pet [3]
##   pet   size   totalpet
##   <chr> <chr>     <dbl>
## 1 cat   small        15
## 2 dog   medium        4
## 3 dog   small         5
## 4 tiger big           1
```

summarise

```
library(tidyverse)
plot <- c(rep(1,2), rep(2,4), rep(3,3))
bird <- c('a','b', 'a','b', 'c', 'd', 'a', 'b', 'c')
area <- c(rep(10,2), rep(5,4), rep(15,3))

birdlist <- data.frame(plot,bird,area)
birdlist
```

```
##   plot bird area
## 1    1    a   10
## 2    1    b   10
## 3    2    a    5
## 4    2    b    5
## 5    2    c    5
## 6    2    d    5
## 7    3    a   15
## 8    3    b   15
## 9    3    c   15
```

```r
# summarize the following data frame to a summary table.
# option 1
birdlist %>%
  group_by(plot) %>%
  summarise(bird = n(), area = unique(area))
```

```
## # A tibble: 3 x 3
##    plot  bird  area
##   <dbl> <int> <dbl>
## 1     1     2    10
## 2     2     4     5
## 3     3     3    15
```

```r
# option 2
birdlist %>%
  count(plot, area, name = "bird")
```

```
##   plot area bird
## 1    1   10    2
## 2    2    5    4
## 3    3   15    3
```

```r
gapminder %>%
  summarise(mean(lifeExp))
```

```
## # A tibble: 1 x 1
##   `mean(lifeExp)`
##             <dbl>
## 1            59.5
```

```r
gapminder %>%
  summarise(range(lifeExp))
```

```
## # A tibble: 2 x 1
##   `range(lifeExp)`
##              <dbl>
## 1             23.6
## 2             82.6
```

```r
gapminder %>%
  filter(country=="India") %>%
  group_by(country) %>%
  summarise(GDPmax=max(gdpPercap),
            GDPmin=min(gdpPercap),
            GDPmean=mean(gdpPercap))
```

```
## # A tibble: 1 x 4
##   country GDPmax GDPmin GDPmean
##   <fct>    <dbl>  <dbl>   <dbl>
## 1 India    2452.   547.   1057.
```

count/summarize

```
library(tidyverse)
plot <- c(rep(1,2), rep(2,4), rep(3,3))
bird <- as.factor(c('a','b', 'a','b', 'c', 'd', 'a', 'b', 'c'))
area <- c(rep(10,2), rep(5,4), rep(15,3))

birdlist <- data.frame(plot,bird,area)
birdlist
```

```
##   plot bird area
## 1    1    a   10
## 2    1    b   10
## 3    2    a    5
## 4    2    b    5
## 5    2    c    5
## 6    2    d    5
## 7    3    a   15
## 8    3    b   15
## 9    3    c   15
```

```
#birdlist %>%    group_by(plot, area) %>%    mutate(count(bird))


birdlist %>%
  group_by(plot, area) %>%
  dplyr::summarize(bird = n(), # when summarize doesn't work directly use it (dplyr::)like this
            .groups = "drop") # to summarize of a column with reference to two other variables.
```

```
## # A tibble: 3 x 3
##    plot  area  bird
##   <dbl> <dbl> <int>
## 1     1    10     2
## 2     2     5     4
## 3     3    15     3
```

count sites

```
treatment <- c(rep('ab',2), rep('bgrnf', 8), rep('bgpnf', 4))
site <- c('ab1', 'ab2',
          rep('bgrnf1', 3),
          rep('bgrnf2', 2),
          'bgrnf3',
          'bgrnf4',
          'bgrnf5',
          rep('bgpnf1', 2),
          rep('bgpnf2', 2))
data <- data.frame(treatment, site)
library(tidyverse)
```

```
# to find the site per each treatment
data %>% group_by(treatment) %>% count(treatment)
```

```
## # A tibble: 3 x 2
## # Groups:   treatment [3]
##   treatment     n
##   <chr>     <int>
## 1 ab            2
## 2 bgpnf         4
## 3 bgrnf         8
```

case when new column

```
library(dplyr)
library(stringr)
feedback <- c('good_book', 'good_read', 'good_story', 'good for knowledge')
book <- c('ramayana', 'bible', 'encyclopedia', 'Mbharatha')

df <- data.frame(feedback, book)

df %>%
  mutate(response = case_when(str_starts(feedback, 'good') ~ 'good')) %>%
  select(book, response) %>% as_tibble()
```

```
## # A tibble: 4 x 2
##   book         response
##   <chr>        <chr>
## 1 ramayana     good
## 2 bible        good
## 3 encyclopedia good
## 4 Mbharatha    good
```

separate

text to columns

```
df <- data.frame(films = c("Spider_man", "James_bond", "Iron_man", "Bat_man"))
df
```

```
##        films
## 1 Spider_man
## 2 James_bond
## 3   Iron_man
## 4    Bat_man
```

```
df1 <- df %>%
  separate(films, c("a", "b"), sep='([_])')
df1
```

```
##        a    b
## 1 Spider  man
## 2  James bond
## 3   Iron  man
## 4    Bat  man
```

unite

```
df1 %>% unite("names", a:b, remove=FALSE)
```

```
##        names      a    b
## 1 Spider_man Spider  man
## 2 James_bond  James bond
## 3   Iron_man   Iron  man
## 4    Bat_man    Bat  man
```

join

```
df2 <- data.frame(id=c(1:4) ,films = c("Spider_man", "James_bond", "Iron_man", "Bat_man"))
df3 <- data.frame(id=c(1:4) ,country= rep("us", 4))

df4 <- left_join(df2, df3, by="id")
df4
```

```
##   id      films country
## 1  1 Spider_man      us
## 2  2 James_bond      us
## 3  3   Iron_man      us
## 4  4    Bat_man      us
```

across

for multiple variables

```
library(tidyverse)
srno <- c(1:2)
film <- c("arabica", "robust")
rate <- c("good", "better")
lang_Eng <- c("yes", "yes")

films <- data.frame(srno, film, rate, lang_Eng)

str(films)
```

```
## 'data.frame':    2 obs. of  4 variables:
##  $ srno    : int  1 2
```

```
##  $ film    : chr  "arabica" "robust"
##  $ rate    : chr  "good" "better"
##  $ lang_Eng: chr  "yes" "yes"
```

```
films <- films %>%
  mutate(across(c(rate, lang_Eng), as.factor))

str(films)
```

```
## 'data.frame':    2 obs. of  4 variables:
##  $ srno    : int  1 2
##  $ film    : chr  "arabica" "robust"
##  $ rate    : Factor w/ 2 levels "better","good": 2 1
##  $ lang_Eng: Factor w/ 1 level "yes": 1 1
```

everthing

select a key variable and everything

```
library(gapminder)
gapminder %>% select(pop, everything())
```

```
## # A tibble: 1,704 x 6
##         pop country     continent  year lifeExp gdpPercap
##       <int> <fct>       <fct>     <int>   <dbl>     <dbl>
##  1  8425333 Afghanistan Asia       1952    28.8      779.
##  2  9240934 Afghanistan Asia       1957    30.3      821.
##  3 10267083 Afghanistan Asia       1962    32.0      853.
##  4 11537966 Afghanistan Asia       1967    34.0      836.
##  5 13079460 Afghanistan Asia       1972    36.1      740.
##  6 14880372 Afghanistan Asia       1977    38.4      786.
##  7 12881816 Afghanistan Asia       1982    39.9      978.
##  8 13867957 Afghanistan Asia       1987    40.8      852.
##  9 16317921 Afghanistan Asia       1992    41.7      649.
## 10 22227415 Afghanistan Asia       1997    41.8      635.
## # ... with 1,694 more rows
```

toupper

tolower

```
library(stringr)

data <- data.frame(Dose.Cm=c("d1", "D2", "D3"),
                   Len.km=c("High", 'low', 'Low'))
glimpse(data)
```

```
## Rows: 3
## Columns: 2
## $ Dose.Cm <chr> "d1", "D2", "D3"
## $ Len.km  <chr> "High", "low", "Low"
```

```
data %>% mutate(Dose.Cm= tolower(Dose.Cm), Len.km=toupper(Len.km))
```

```
##   Dose.Cm Len.km
## 1      d1   HIGH
## 2      d2    LOW
## 3      d3    LOW
```

factor

```
data <- data.frame(Dose.Cm=c("d1", "D2", "D3"),
                   Len.km=c("high", 'low', 'medium'))
data <- data %>% mutate(len= as.factor(Len.km))

glimpse(data)
```

```
## Rows: 3
## Columns: 3
## $ Dose.Cm <chr> "d1", "D2", "D3"
## $ Len.km  <chr> "high", "low", "medium"
## $ len     <fct> high, low, medium
```

change order of factor

```
data %>% mutate(len= fct_relevel(len, c('low', 'medium', 'high')))
```

```
##   Dose.Cm Len.km    len
## 1      d1   high   high
## 2      D2    low    low
## 3      D3 medium medium
```

parse_number

This drops any non-numeric characters before or after the first number. The grouping mark specified by the locale is ignored inside the number.

```
library(tidyverse)
class <- c('8th', '9th', '10th')
students <- c('25-30', '35-41', '21-28')
school <- data.frame(class, students)
school
```

```
##   class students
## 1   8th    25-30
## 2   9th    35-41
## 3  10th    21-28
```

```
glimpse(school) # notice students is a binned variable it is a not a numeric.
```

```
## Rows: 3
## Columns: 2
## $ class    <chr> "8th", "9th", "10th"
## $ students <chr> "25-30", "35-41", "21-28"
```

```
school %>% mutate(students= parse_number(students)) %>% glimpse()
```

```
## Rows: 3
## Columns: 2
## $ class    <chr> "8th", "9th", "10th"
## $ students <dbl> 25, 35, 21
```

```
school %>% mutate(students= parse_number(students))
```

```
##   class students
## 1   8th       25
## 2   9th       35
## 3  10th       21
```

```
# now students because number with first value of the column
```

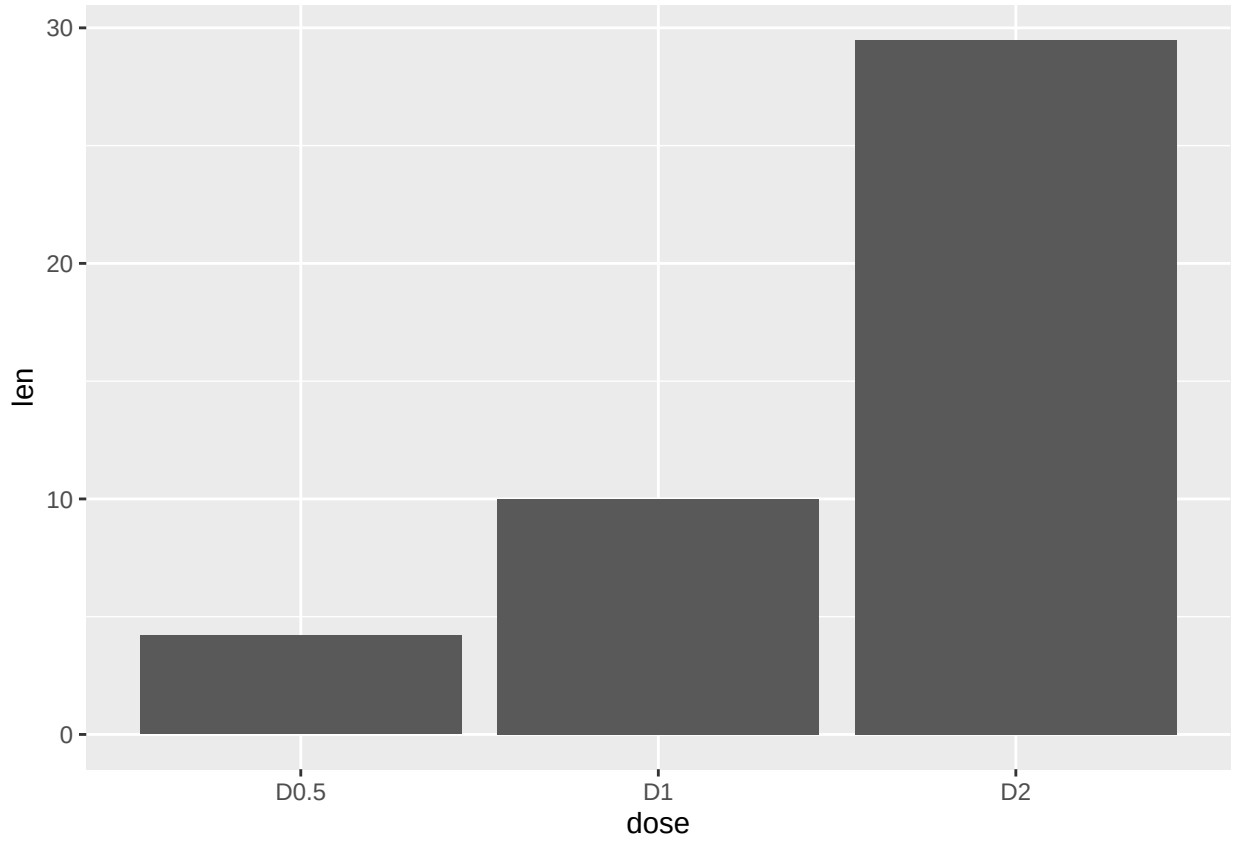
pivot longer

```
library(tidyverse)

rawdata <- data.frame(species_1=rnorm(n = 40, mean = 300, sd = 18.5), species_2=rnorm(40, 305, 16.7))


data <- pivot_longer(data = rawdata, cols = species_1:species_2, names_to = 'species', values_to = 'wei
```

ggplot

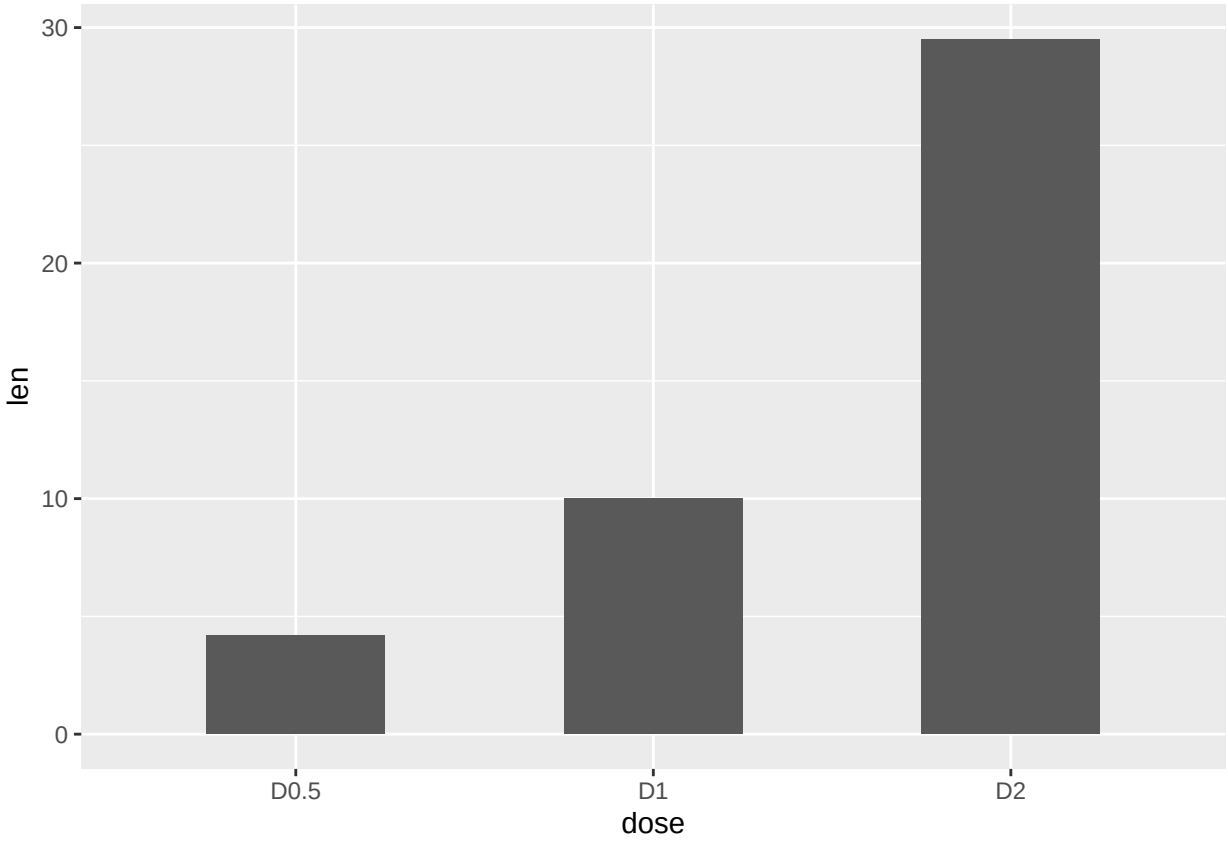
```
#sthda.com/english/wiki/ggplot2-barplots-quick-start-guide-r-software-and-data-visualization
df <- data.frame(dose=c("D0.5", "D1", "D2"),
                 len=c(4.2, 10, 29.5))
```

```
library(ggplot2)
# Basic barplot
p<-ggplot(data=df, aes(x=dose, y=len)) +
  geom_bar(stat="identity")
p
```

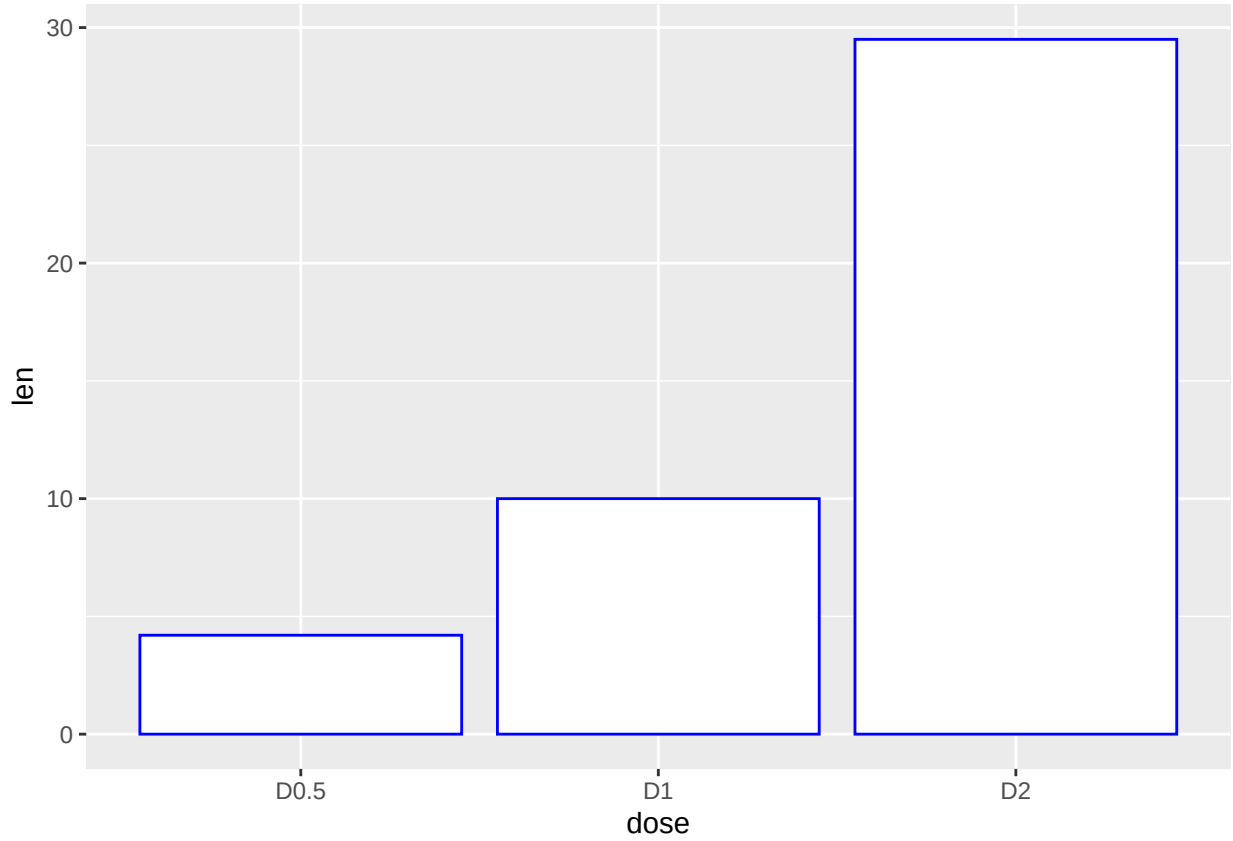



```
# Horizontal bar plot
# p + coord_flip()
```

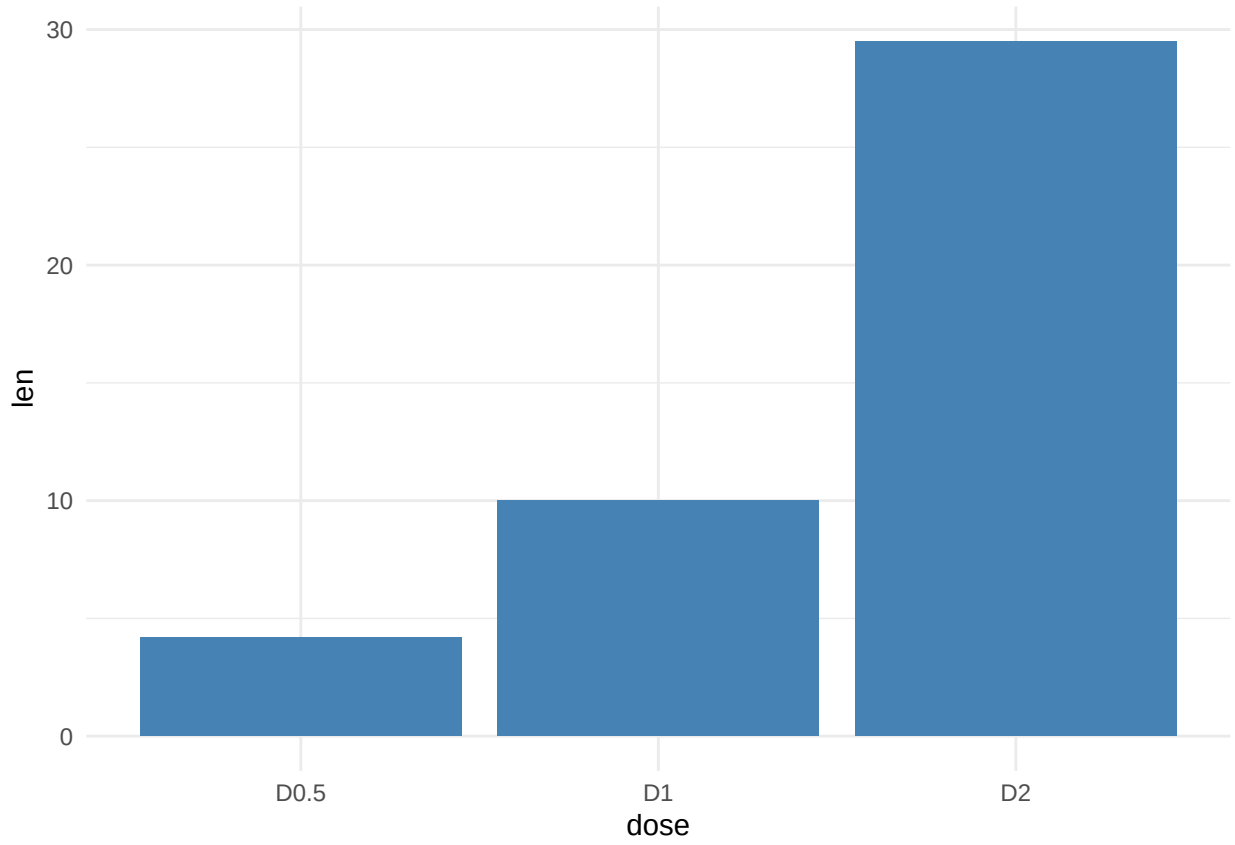
```
# Change the width of bars
ggplot(data=df, aes(x=dose, y=len)) +
  geom_bar(stat="identity", width=0.5)
```




```
# Change colors
ggplot(data=df, aes(x=dose, y=len)) +
  geom_bar(stat="identity", color="blue", fill="white")
```




```
# Minimal theme + blue fill color
p<-ggplot(data=df, aes(x=dose, y=len)) +
  geom_bar(stat="identity", fill="steelblue")+
  theme_minimal()
p
```



months

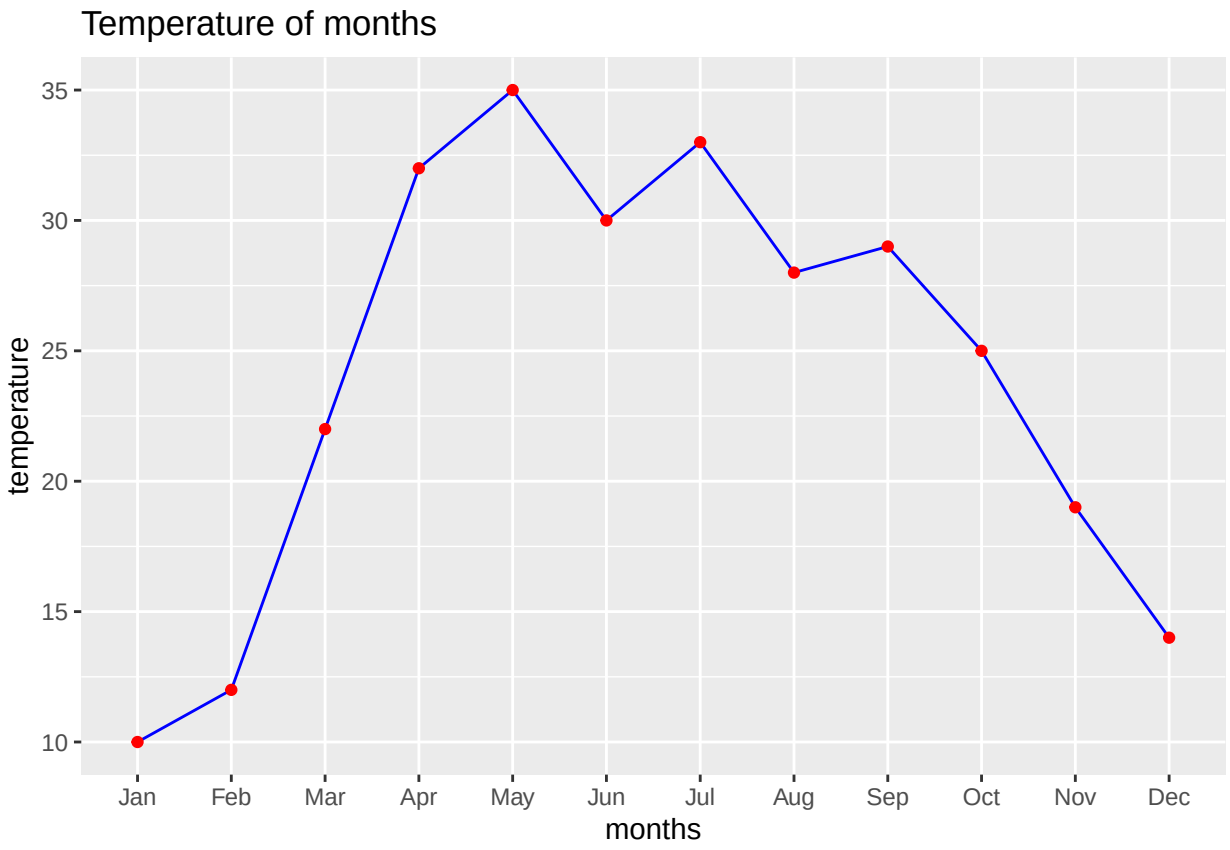
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
months <- seq(month(1:12)) # make moths
months <- month.abb[months] # make abbriviations
temperature <- c(10,12,22,32,35,30,33,28,29,25,19,14)
myframe <- data.frame(months,temperature) # creating a new data frame

library(tidyverse)
glimpse(myframe)
```

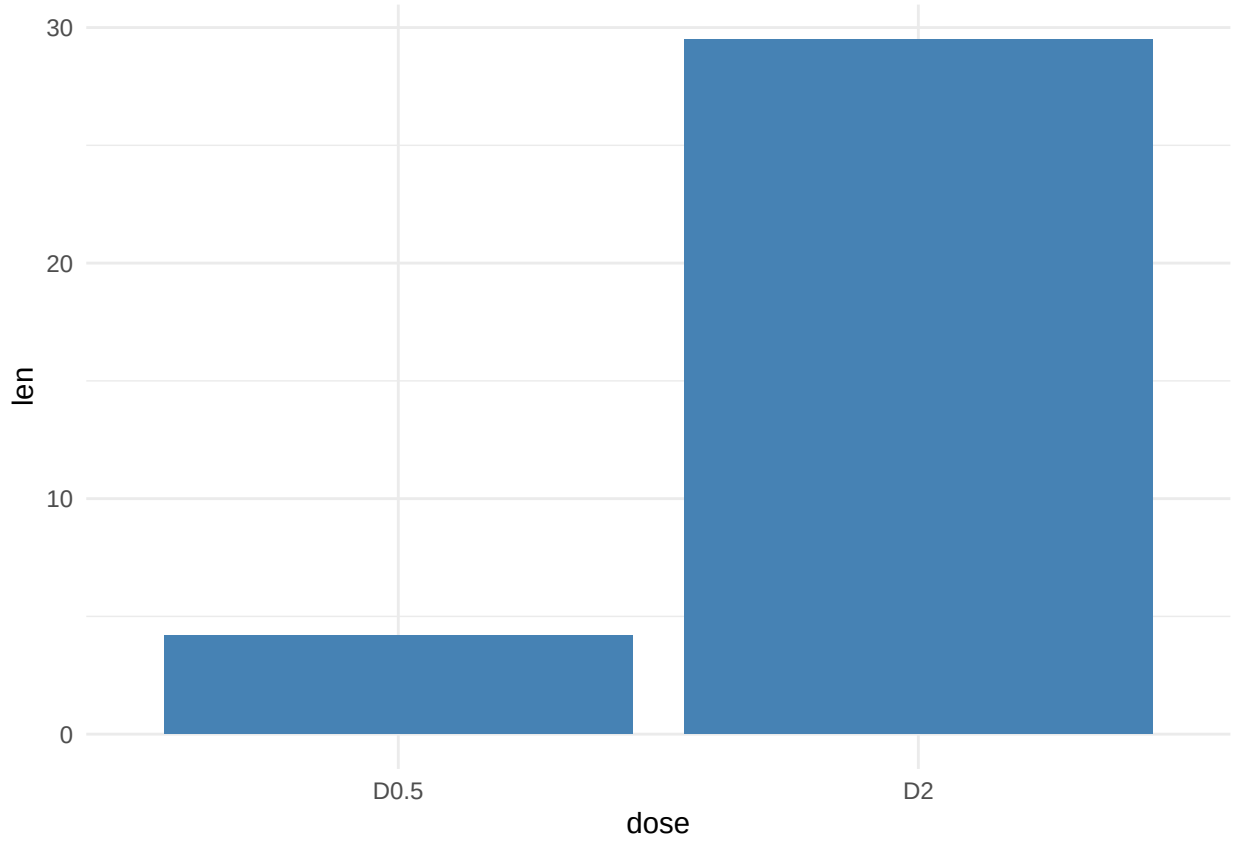
```
## Rows: 12
## Columns: 2
## $ months      <chr> "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "S~
## $ temperature <dbl> 10, 12, 22, 32, 35, 30, 33, 28, 29, 25, 19, 14
```

```
library(ggplot2)
ggplot(myframe, aes(x=months, y=temperature, group=1))+
  geom_line(col='blue')+
  geom_point(col='red')+
  ggtitle('Temperature of months')+
  scale_x_discrete(limits = month.abb) # this will order months on the x axis
```

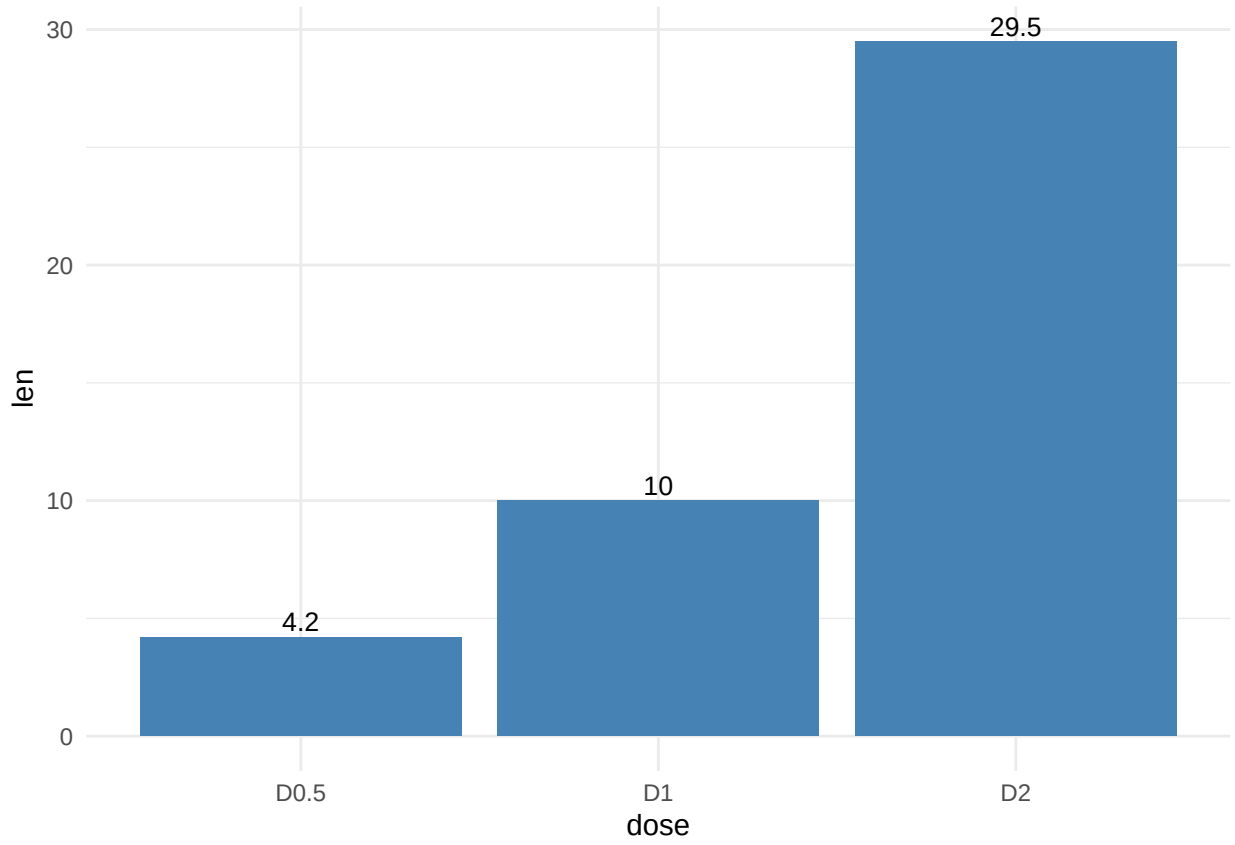



```
p + scale_x_discrete(limits=c("D0.5", "D2"))
```

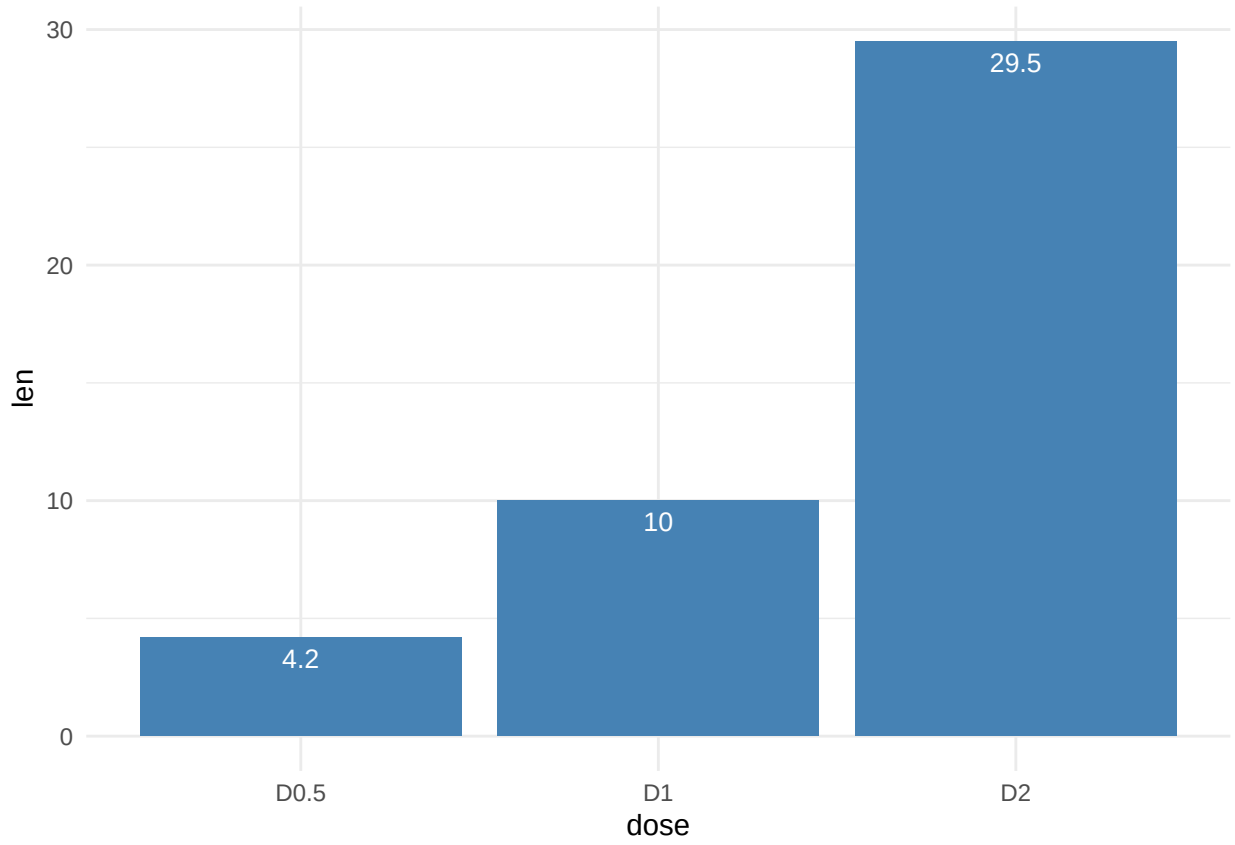
```
## Warning: Removed 1 rows containing missing values (position_stack).
```




```
# Outside bars
ggplot(data=df, aes(x=dose, y=len)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=len), vjust=-0.3, size=3.5)+
  theme_minimal()
```

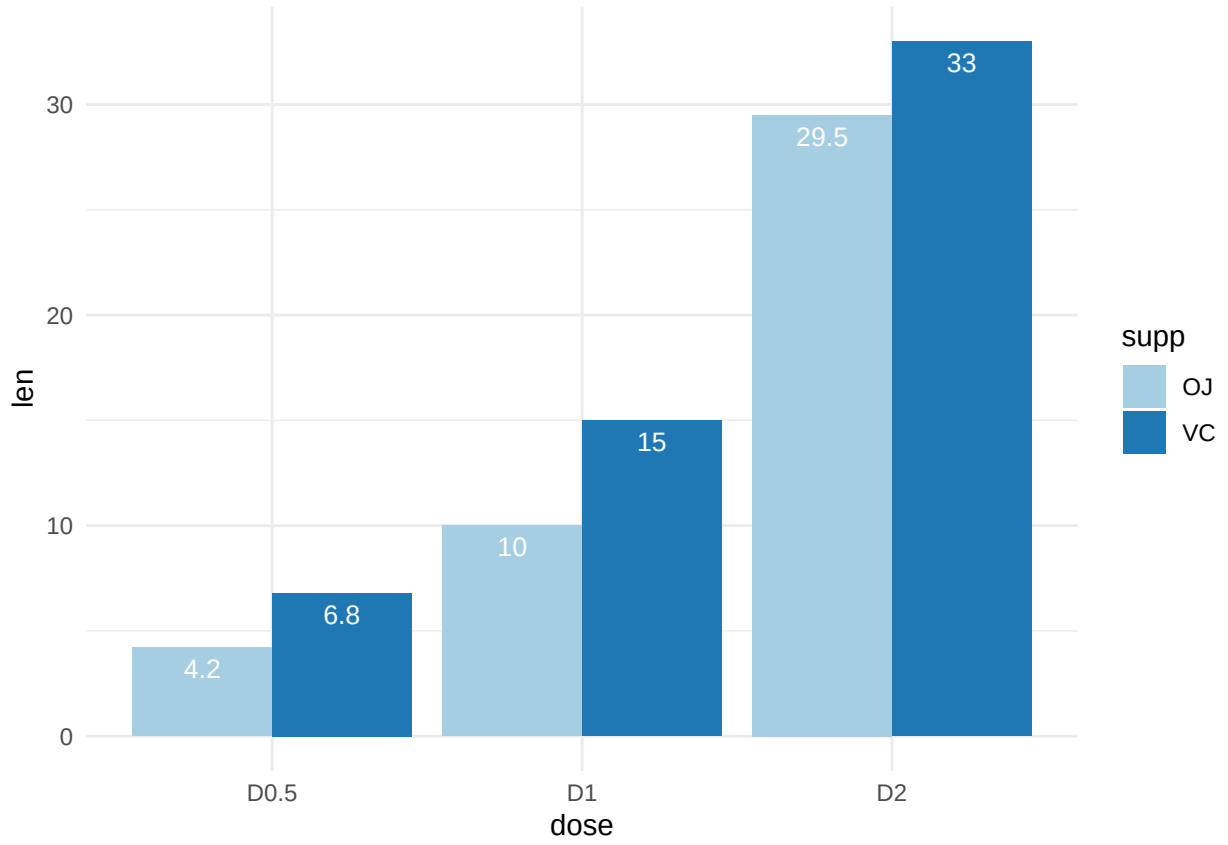



```
# Inside bars
ggplot(data=df, aes(x=dose, y=len)) +
  geom_bar(stat="identity", fill="steelblue")+
  geom_text(aes(label=len), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```

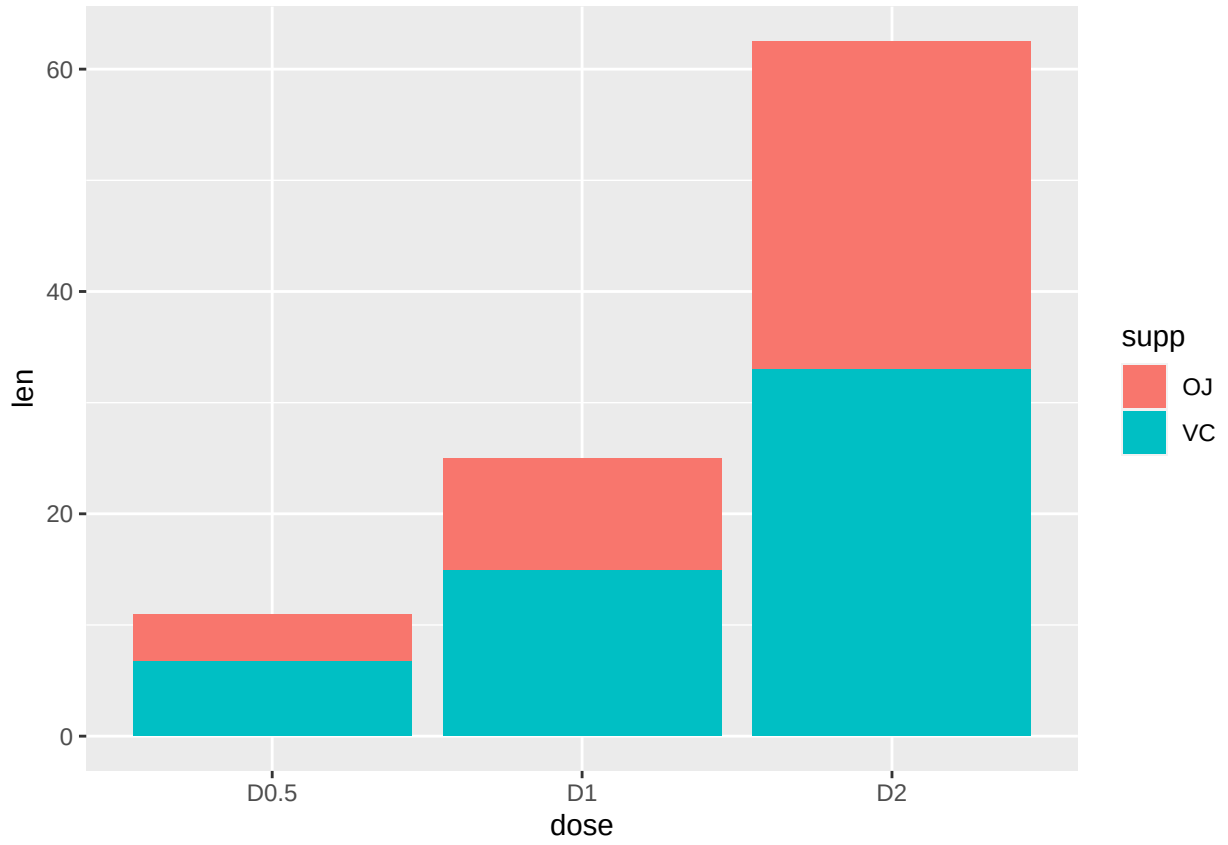



```
df2 <- data.frame(supp=rep(c("VC", "OJ"), each=3),
                  dose=rep(c("D0.5", "D1", "D2"),2),
                  len=c(6.8, 15, 33, 4.2, 10, 29.5))
```

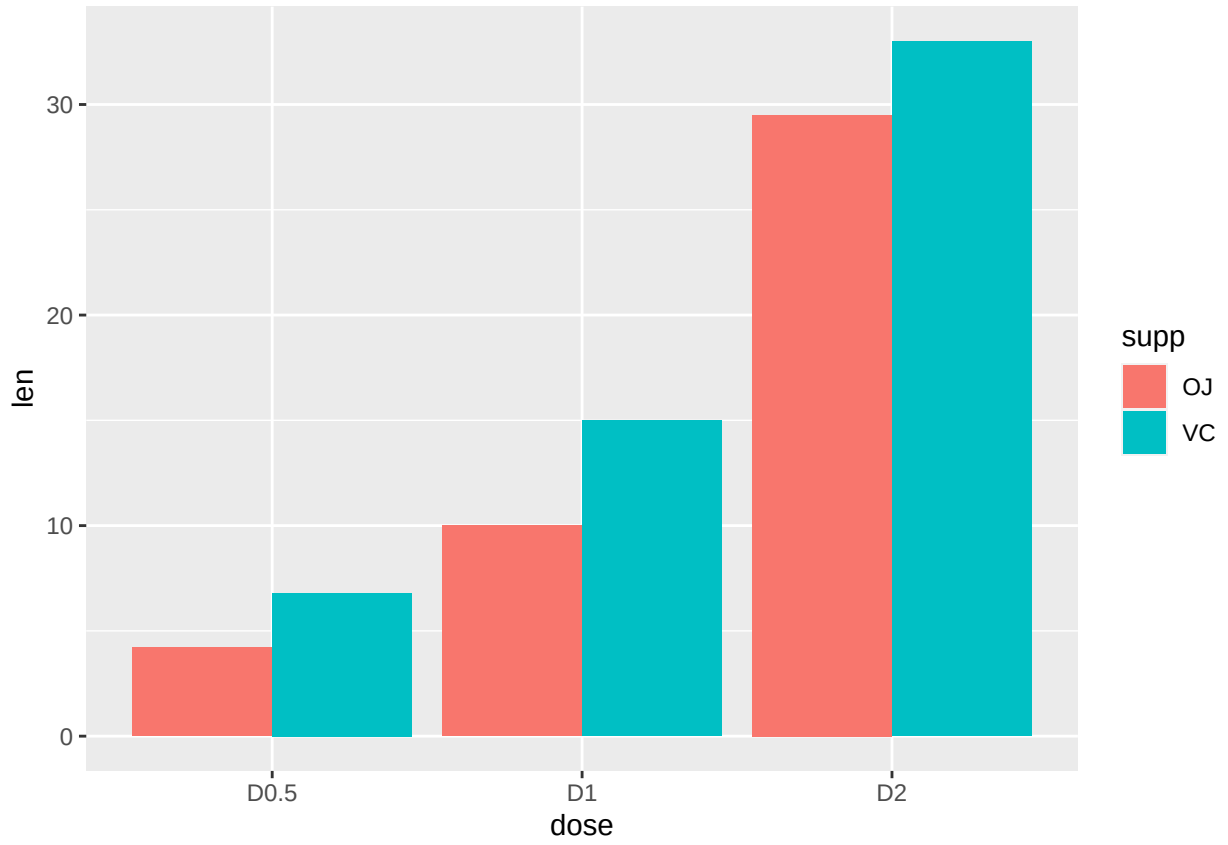
```
ggplot(data=df2, aes(x=dose, y=len, fill=supp)) +
  geom_bar(stat="identity", position=position_dodge())+
  geom_text(aes(label=len), vjust=1.6, color="white",
            position = position_dodge(0.9), size=3.5)+
  scale_fill_brewer(palette="Paired")+
  theme_minimal()
```



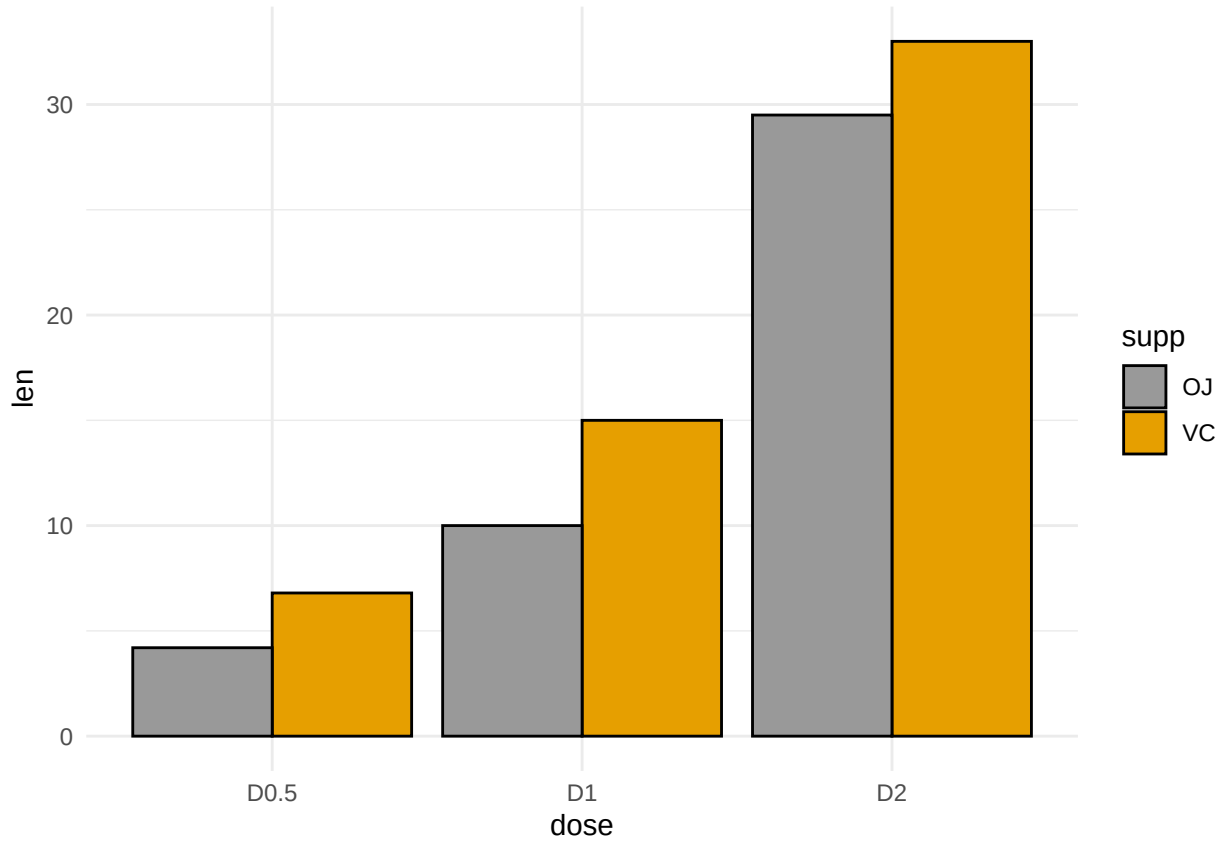

```
# Stacked barplot with multiple groups
ggplot(data=df2, aes(x=dose, y=len, fill=supp)) +
  geom_bar(stat="identity")
```



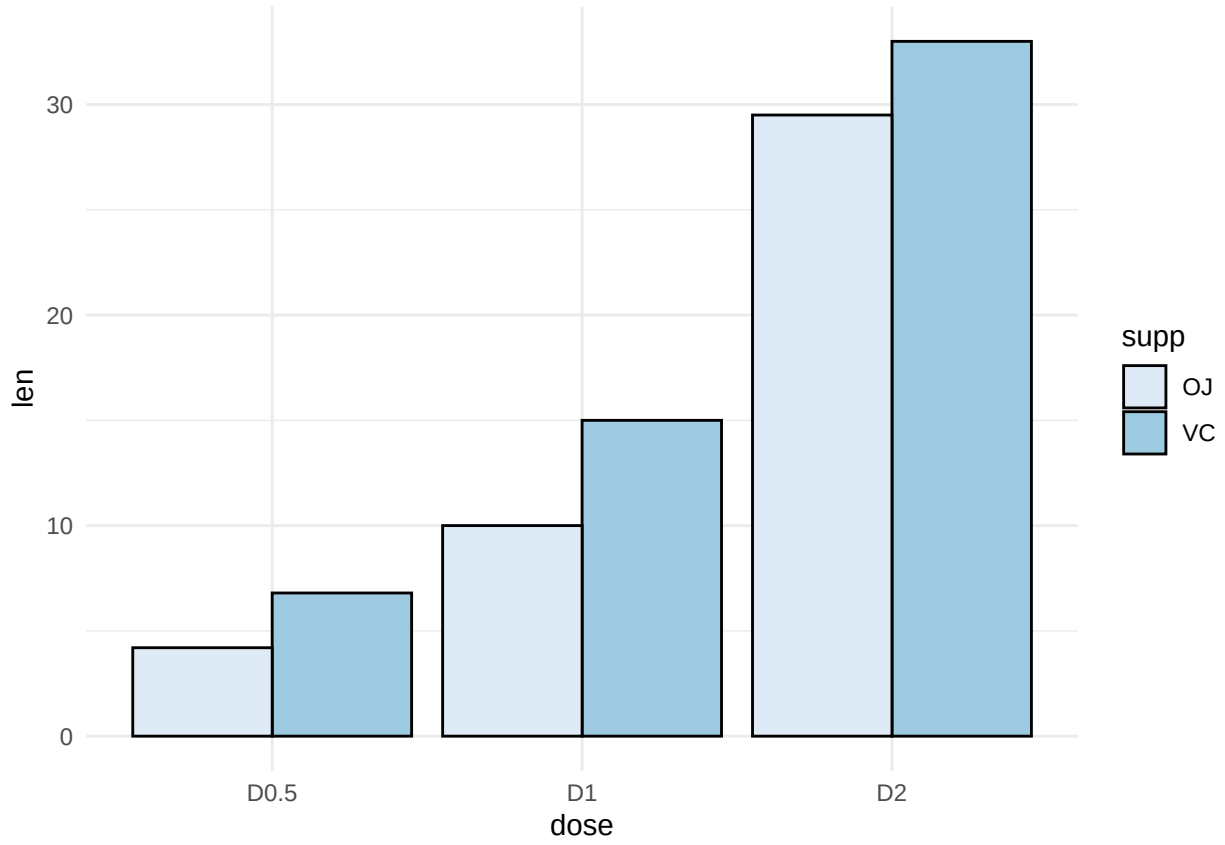

```
# Use position=position_dodge()
ggplot(data=df2, aes(x=dose, y=len, fill=supp)) +
geom_bar(stat="identity", position=position_dodge())
```




```
# Change the colors manually
p <- ggplot(data=df2, aes(x=dose, y=len, fill=supp)) +
geom_bar(stat="identity", color="black", position=position_dodge())+
  theme_minimal()
# Use custom colors
p + scale_fill_manual(values=c('#999999','#E69F00'))
```



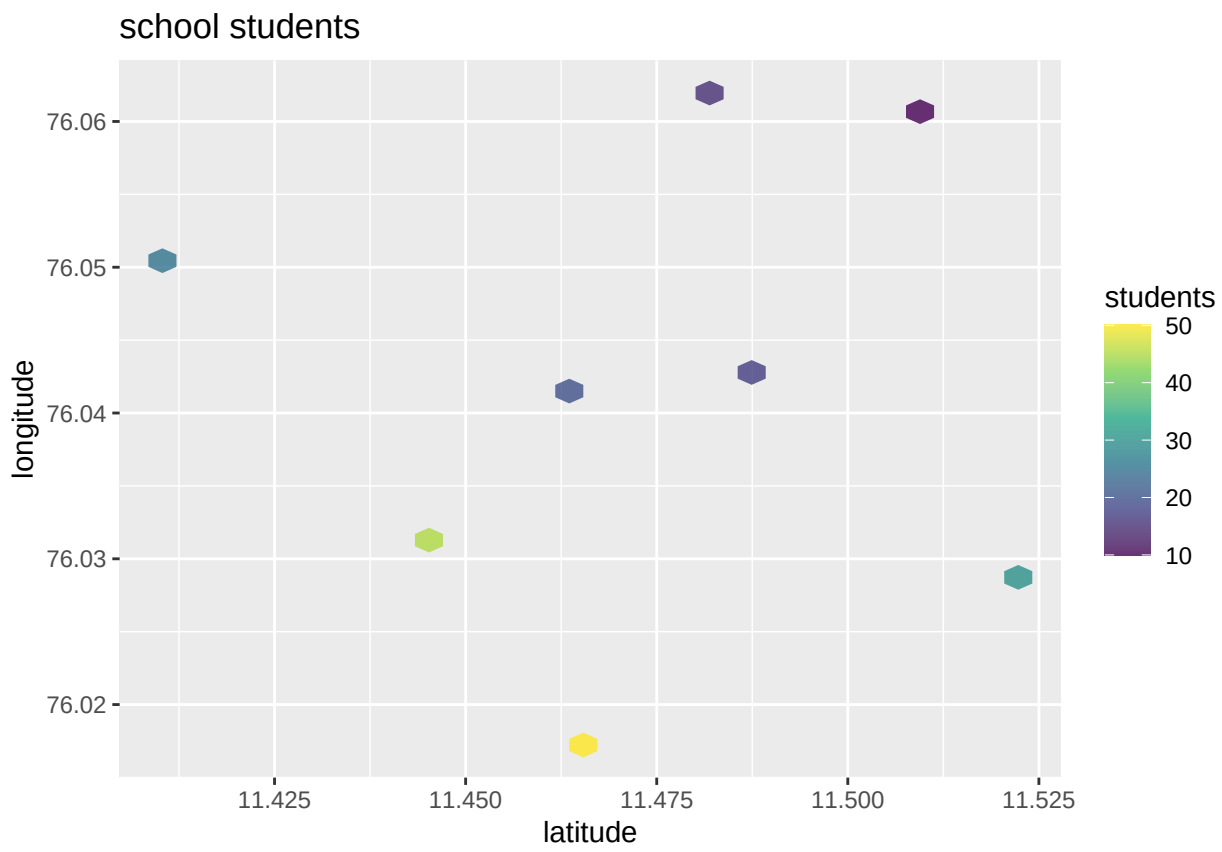

```
# Use brewer color palettes
p + scale_fill_brewer(palette="Blues")
```



hex plot

```
library(tidyverse)
# install.packages("hexbin")
class <- c(rep('10th', 8))
students <- c('10 to 15',
              "15-20",
              "17 to 24",
              "20  to 25",
              "25 to 30",
              "30 to 40",
              "45 to 47",
              '50 to 55')
latitude <- c(11.50897246,
              11.48323136,
              11.48719031,
              11.46366611,
              11.41097322,
              11.52111154,
              11.44491386,
              11.46569568)
longitude <- c(76.06032062,
               76.06192685,
```

```
                76.04266851,
                76.04156575,
                76.05075092,
                76.02846331,
                76.03084141,
                76.01766216)
school <- data.frame(class, students, latitude, longitude)

school %>% mutate(students= parse_number(students)) %>%
  ggplot(aes(latitude, longitude, z= students))+
  stat_summary_hex()+
  scale_fill_viridis_c(alpha= 0.8)+
  labs(fill='students', title = 'school students')
```



Color Palettes

Resources

- https://colorhunt.co/
- https://coolors.co/
- https://colorpalettes.net/
- https://www.canva.com/colors/color-palettes/
- https://color.adobe.com/de/create/color-wheel
- https://mycolor.space/

- http://colormind.io/

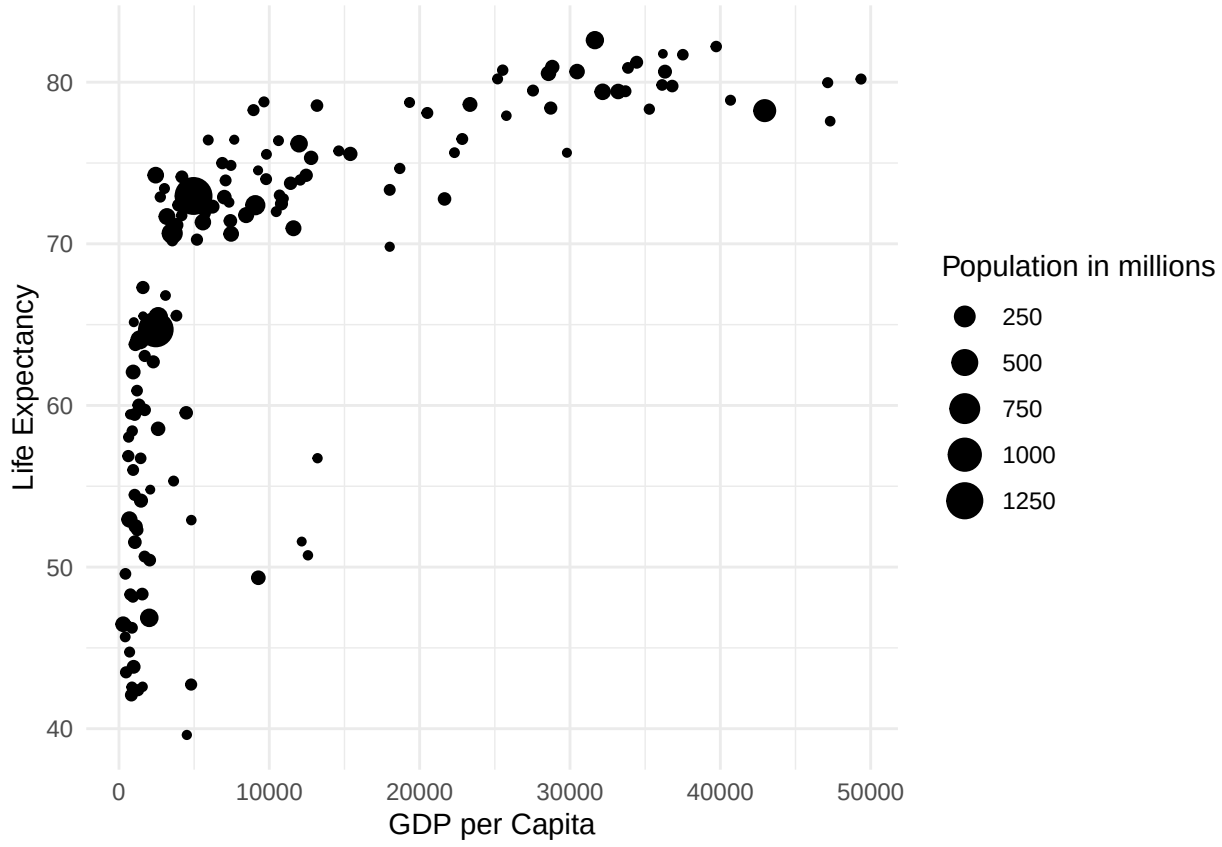
```
#install.packages(c("tidyverse", "gapminder", "MetBrewer"))
```

libraries

```
library(tidyverse)
library(gapminder)
# install.packages('MetBrewer')
library(MetBrewer)
```

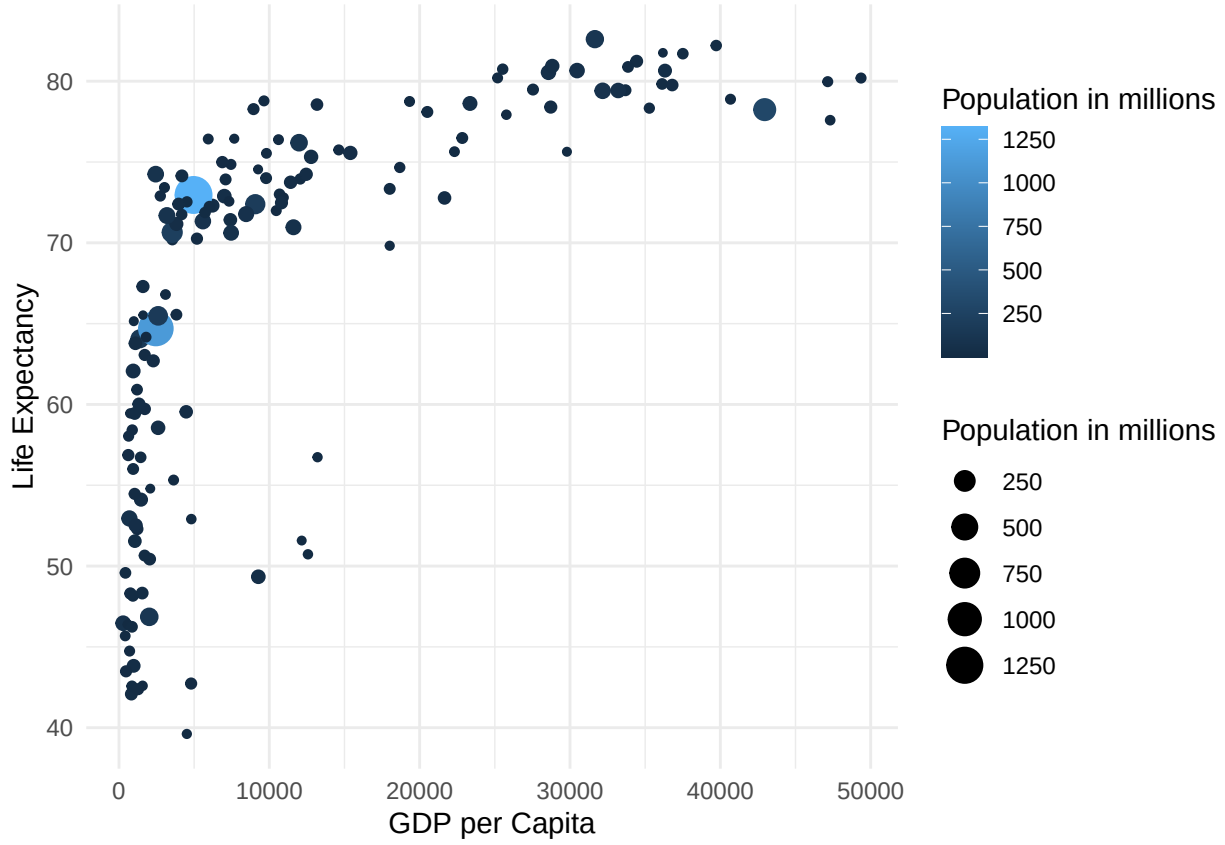
Plot the point plot using GDP per Capita as the x- axis and LE as the y axis. Numerical variable Population to control the size of each point.

```
plot <- gapminder %>%
  filter (year==2007) %>%
  ggplot()+
  labs(x= 'GDP per Capita',
              y= 'Life Expectancy',
       color= 'Population in millions',
       size='Population in millions')+
  theme_minimal()

plot+ geom_point(aes(gdpPercap, lifeExp, size= pop/1000000))
```



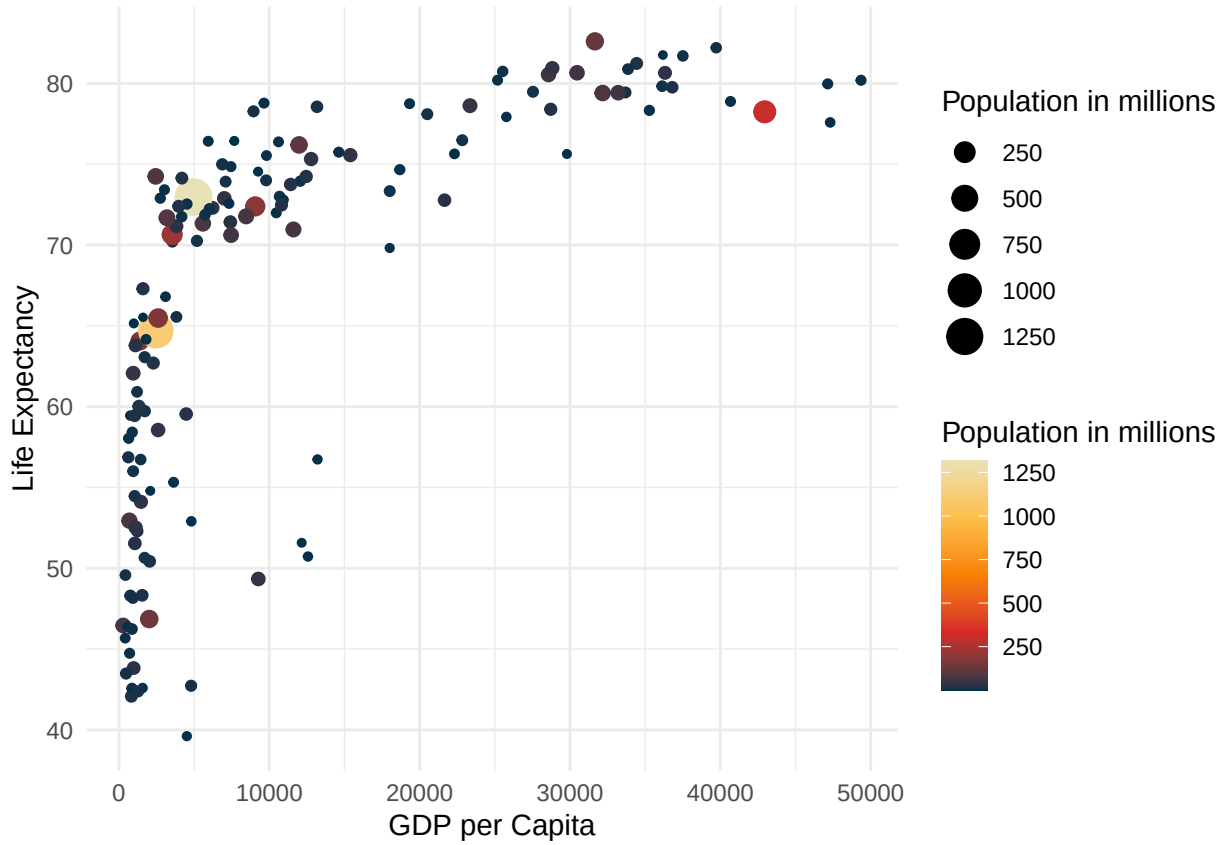
To use color in the plot, assign the Population variable to the color aesthetic. Since nothing is specied, ggplot2 chooses a color spectrum for this numerical variable (shades of blue).

```
plot + geom_point(aes(gdpPercap, lifeExp, size= pop/1000000, color= pop/1000000))
```



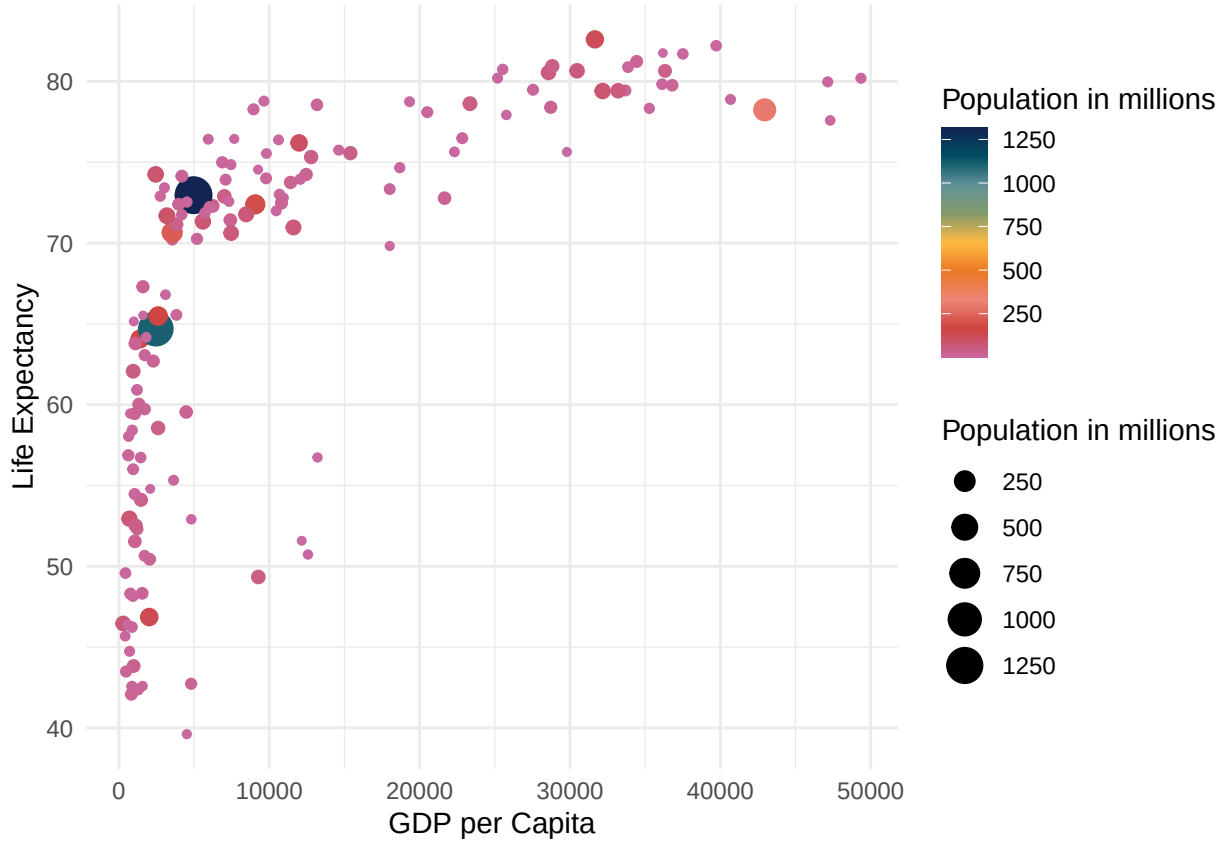
To control the color spectrum, we need to introduce a color scale. In the following plot, we have to provide a vector of hex color values. You would choose this if you got your colors from one of the mentioned above websites.

```
plot + geom_point(aes(gdpPercap, lifeExp, size= pop/1000000, color= pop/1000000))+
  scale_color_gradientn(colors = c("#003049", "#D62828", "#F77F00", "#FCBF49", "#EAE2B7"))
```



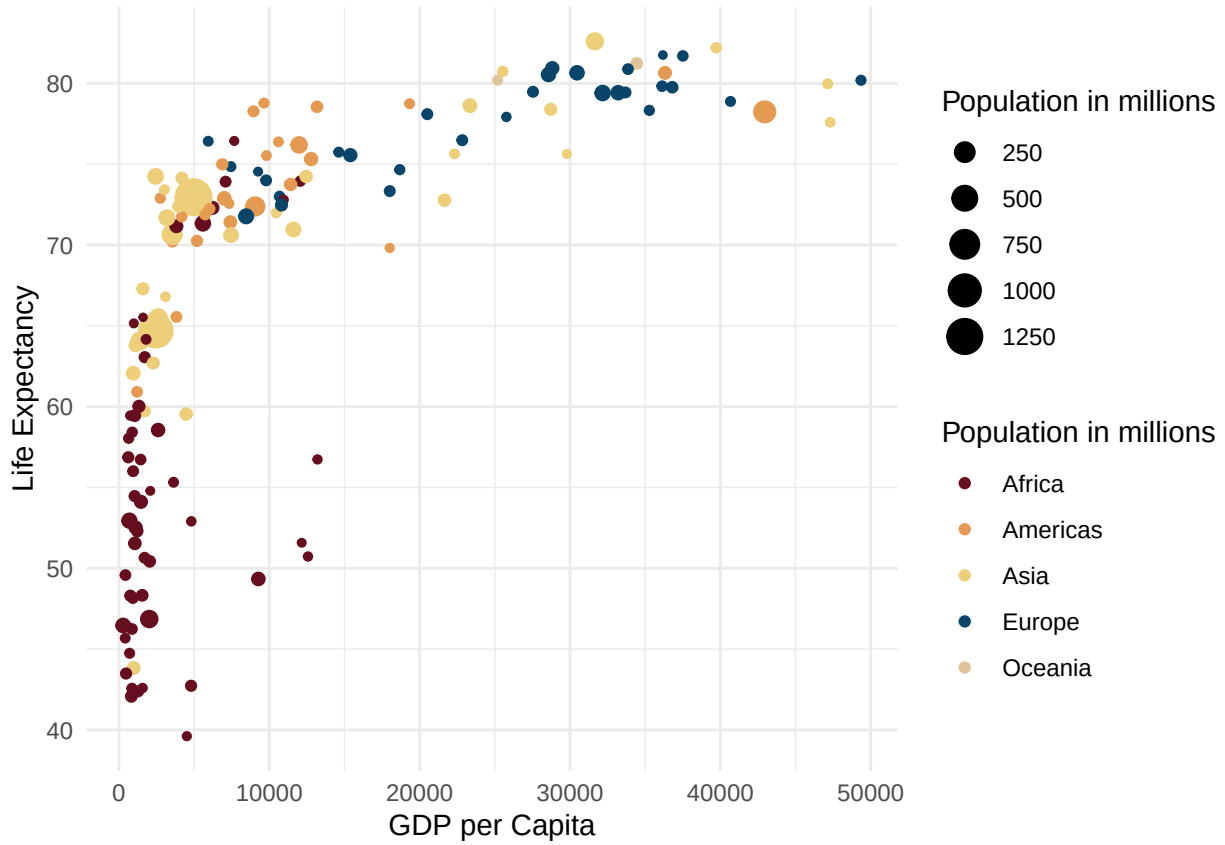
To apply one of the MetBrewer palettes, replace the hex-vector with a MetBrewer function. Within the function call, you provide the palette's name, then several colors, and tell it that we need a continuous palette since it is a numerical variable.

```
plot + geom_point(aes(gdpPercap, lifeExp, size= pop/1000000, color= pop/1000000))+
  scale_color_gradientn(colors = met.brewer('Cross', n=500, type = 'continuous'))
```



You might also want to use color palettes with non-numerical variables. Let us assume we want to apply color to the Continent variable. This implies using a manual color scale and providing a MetBrewer palette.

```
plot + geom_point(aes(gdpPercap, lifeExp, size= pop/1000000, color= continent))+
  scale_color_manual(values = met.brewer('Navajo', 5))
```



Please note if you want to apply color to the fill aesthetic rather than the color aesthetic, consider using the scale_fill_manuel function instead of the scale_color_manuel. This is useful for boxplots or bar charts.

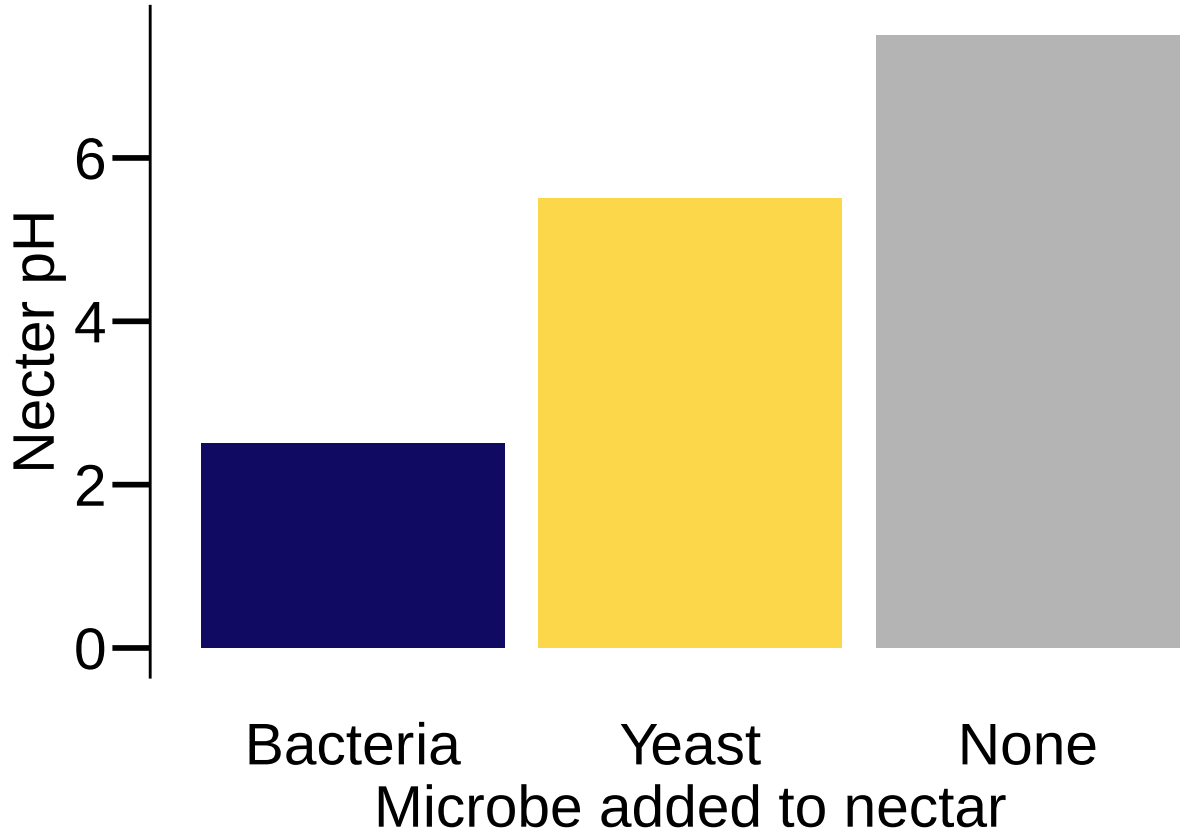
```
box <- gapminder %>%
  filter(gdpPercap< 60000) %>%
  ggplot(aes(continent, gdpPercap, color= year, fill= continent))+
           geom_boxplot()+
  theme_minimal()+ labs( x= 'Continent', y= 'GDP per Capita', fill= 'Continent')
```

scale fill manual

themes

```
df <- data.frame(
    Names=as.factor(c('Bacteria', 'Yeast', 'None')),
    Quantity=c(2.5, 5.5, 7.5))

library(ggplot2)
library(tidyverse)
df <- df %>% mutate(Names= fct_relevel(Names, c('Bacteria', 'Yeast', 'None')))

ggplot(df, aes(Names, Quantity, fill= Names))+
```

```
    geom_bar(stat = 'identity')+
    scale_fill_manual(values = c('#110a62', '#fcd749','#b5b4b5'))+
  labs(y='Necter pH', x= 'Microbe added to nectar')+
  theme_classic()+
    theme(legend.position = 'none', axis.ticks.x = element_blank())+
  theme(axis.text = element_text(size = 22, color= 'black'))+
theme(axis.line.x = element_blank())+
  theme(axis.ticks = element_line(size = 1, color="black"),
   axis.ticks.length = unit(.5, "cm"))+
  theme(text = element_text(size = 22))
```



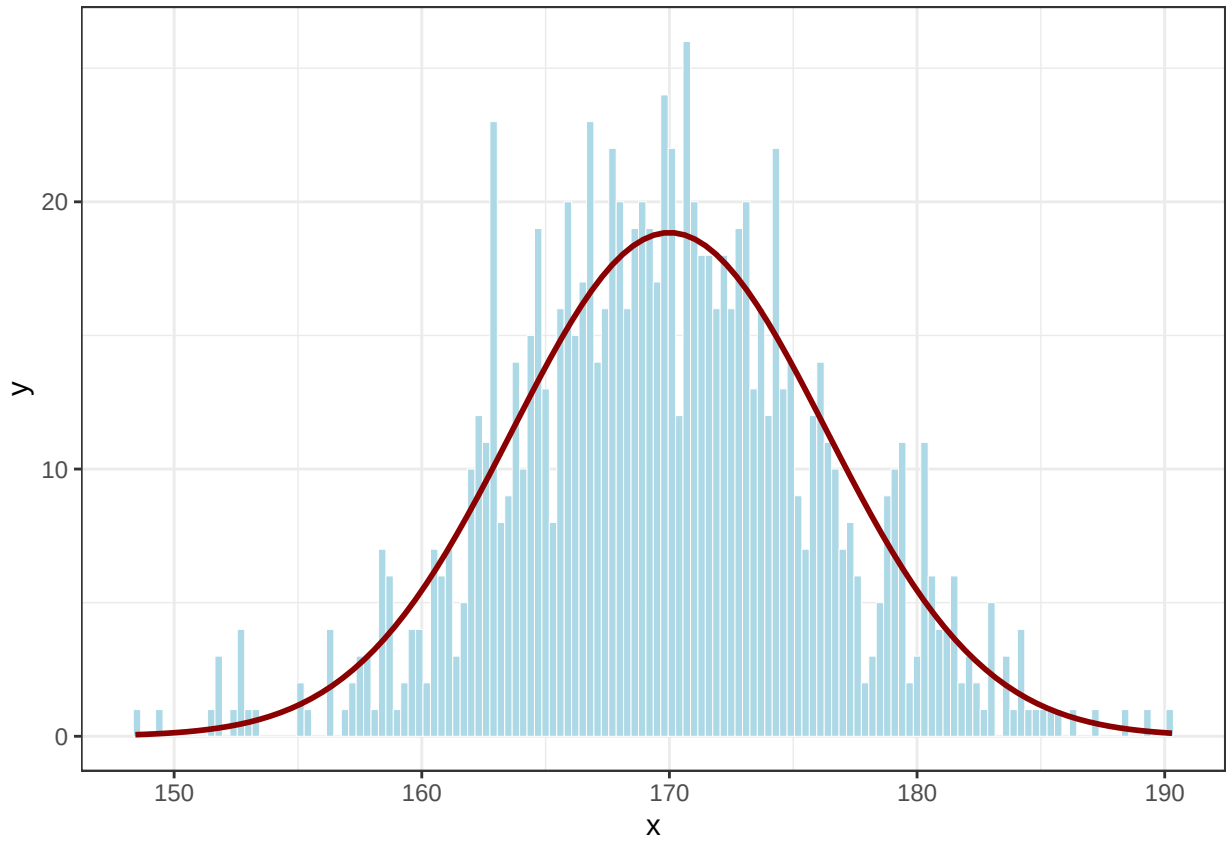
graphics

```
x11() # opne a new window for graphics
graphics.off() # close the new window
```

Normal distribution

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean.

```
library(tidyverse)
n = 1000
mean = 170 # cm
sd = 6.35 # cm
binwidth= 0.3
set.seed(1234)
df <- data.frame(x=rnorm(n, mean, sd))
ggplot(df, aes(x = x, mean = mean, sd = sd, binwidth = binwidth, n = n))+
    theme_bw()+
  geom_histogram(binwidth = binwidth,
        colour = "white", fill = "lightblue", size = 0.1)+
stat_function(fun = function(x) dnorm(x, mean = mean, sd = sd) * n * binwidth,
    color = "darkred", size = 1)
```



Functions

dice

```
dice <- c(1:6)

myluck<- function(x){
  myluck <- sample(dice, size = 1, replace = T)
```

```
  return(myluck)
}

myluck()
```

```
## [1] 2
```

pick a name

```
names <- c('saneesh','appu','sanusha')
who <- function(x){
  who <- sample(names, 1, T)
  return(who)
}

who()
```

```
## [1] "saneesh"
```

function to split

```
df <- data.frame(name=as.factor(c('James Bond', 'Spider Man', 'Iron Man')))
# df <- df %>% separate(name, c('Genus', 'Species'), sep = '([ ])')

shorten <- function(df){
  name_split <- df %>% separate(name, c('Genus', 'Species'), sep = '([ ])')
  print(name_split)
}

shorten(df)
```

```
##    Genus Species
## 1  James    Bond
## 2 Spider     Man
## 3   Iron     Man
```