A book for practicing R

Daryn Ramsden

2019-08-03

Contents

Chapter 1

Intro

This book/site is written to accompany an introductory workshop covering:

- the R programming language
- the R software application
- the RStudio software application.

Software Installation

R and **RStudio** can be downloaded from the following sites:

- Download R @ The R Project's Home Page
  - Windows
  - Linux
  - Mac
- Download RStudio Desktop

After downloading, you should install **R** and **RStudio** in that order.

One additional R package

This workshop requires one additional package. You can install it by opening **R** and running the following commands. (This package contains some data that we will be using.)

```
install.packages("devtools")
library(devtools)
install_github("thisisdaryn/notitia")
```

If your installation is successful, you should be able to run the commands below.

```
library(notitia)
populations
```

```
## # A tibble: 8 x 2
##   country         pop
##   <chr>         <dbl>
## 1 India          1311
## 2 United States   331
## 3 Indonesia       264
## 4 Pakistan        210
## 5 Nigeria         208
## 6 Bangladesh      161
## 7 Russia          141
## 8 Mexico          127
```

Chapter 2

The R Console

The R console is an interactive environment. The user can enter commands/statements and submit them to be run by the computer.

Using R as a calculator

The most basic statements we can use the console for are for using R as a calculator. The commands below are examples. After each command is submitted and run, R will return an appropriate answer.

```
3*4
```

```
## [1] 12
```

```
99 - 1001
```

```
## [1] -902
```

```
4^2
```

```
## [1] 16
```

<- the assignment operator

The first operator we will look at is <-, the assignment operator. Type in the commands below to verify the output.

```
var1 <- 99
var1
```

```
## [1] 99
```

Using, **<-** had the effect of associating the value on its right side with the name on its left side. This command created a **variable**. To see the value of the variable that has been created you can type the variable name at the console.

We can create multiple variables and use them in calculations

```
var2 <- -10001.99
var1*var2
```

```
## [1] -990197
```

We can also use **<-** to change the value of a variable that was previously computed.

```
var1 <- 45
var1
```

```
## [1] 45
```

At any given time, commands that are run by R that are using variables will use the current value of the variable

```
var1*var2
```

```
## [1] -450089.5
```

Chapter 3

Data in R

Types of data

- numeric
- integer
- character
- logical
- factor
- obscure types that you may never encounter
  - complex
  - raw

Common data structures in R

- Very common Data Structures
  - atomic vector: a one-dimensional list of values all of the same type
  - data frame: a collection of atomic vectors all of the same length. Corresponds to a table of data
- Other built-in data structures
  - list: a one-dimensional list of values that are not necessarily of the same type
  - matrix: a two-dimensional collection of values all of the same type

Atomic vectors

A vector is a collection of values all of the same type. In many other languages this . (In Python, this would be used as a list).

```
vec1 <- c(1, 2.5, 1729, -1, 2001)
vec1
```

```
## [1]    1.0    2.5 1729.0   -1.0 2001.0
```

```
class(vec1)
```

```
## [1] "numeric"
```

```
vec2 <- c(FALSE, FALSE, TRUE)
vec3 <- c("A", "collection", "of", "words")
vec4 <- c(1L, 2L, 4L, 8L)
```

```
class(vec4)
```

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

Data frames

A data frame is a group of atomic vectors each of the same length. A data frame corresponds to data in a tabular form. Each of the vectors that comprise the data frame represent one column of the table.

Data frames can be made using the **data.frame** command.

```
country <- c("Algeria", "Barbados", "Cameroon", "Djibouti", "Eritrea")
capital <- c("Algiers", "Bridgetown", "Yaounde", "Djibouti City", "Asmara")
population <- c(42713853, 285719, 25342766, 956985, 5315509)

df <- data.frame(country, capital, population)
df
```

```
##    country       capital population
## 1  Algeria       Algiers   42713853
## 2 Barbados    Bridgetown     285719
## 3 Cameroon       Yaounde   25342766
## 4 Djibouti Djibouti City     956985
## 5  Eritrea        Asmara    5315509
```

However this is often an impractical means of entering data and data frames are typically read in from files or other sources.

Lists

Matrices

A matrix is a two-dimensional data structure in which all the elements are of the same atomic type.

```
A = matrix(
  c("upper left", "upper middle", "upper right", "lower left", "lower middle", "lower right"), #
  nrow=2,               # number of rows
  ncol=3,               # number of columns
  byrow = TRUE)

A
```

```
##      [,1]         [,2]           [,3]
## [1,] "upper left" "upper middle" "upper right"
## [2,] "lower left" "lower middle" "lower right"
```

```
class(A)
```

```
## [1] "matrix"
```

Chapter 4

Operators in R

Logical Operators

!

&

|

&&

||

Relational Operators

<

>

<=

>=

==

!=

Chapter 5

Reading data from files

Reading .csv files

Using read.csv

One of the most commonly-used R commands for reading in data is the **read.csv** function that is built into R. Below is an example:

```
df <- read.csv("data/life-expectancy.csv", stringsAsFactors = FALSE)
```

Using read_csv from the readr package

```
library(readr)
df2 <- read_csv("data/life-expectancy.csv")
```

```
df2
```

```
## # A tibble: 17,894 x 4
##    Entity     Code   Year `Life expectancy (Clio-Infra up to 1949; UN Popu~
##    <chr>      <chr> <dbl>                                              <dbl>
##  1 Afghanist~ AFG    1950                                               27.5
##  2 Afghanist~ AFG    1951                                               27.8
##  3 Afghanist~ AFG    1952                                               28.4
##  4 Afghanist~ AFG    1953                                               28.9
##  5 Afghanist~ AFG    1954                                               29.4
##  6 Afghanist~ AFG    1955                                               29.9
```

```
##  7 Afghanist~ AFG    1956                                                30.4
##  8 Afghanist~ AFG    1957                                                30.9
##  9 Afghanist~ AFG    1958                                                31.4
## 10 Afghanist~ AFG    1959                                                31.8
## # ... with 17,884 more rows
```

Differences between read.csv and read_csv

One difference between the two functions is indicated by using the **class** function on both.

```
class(df)
```

```
## [1] "data.frame"
```

```
class(df2)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Reading in Excel spreadsheets

5.0.1 Using read_excel from the readxl package

```
library(readxl)
nyc_flights <- read_excel("data/NYC_Flights_2013.xlsx", sheet = "Flights")
head(nyc_flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time
##   <dbl> <dbl> <dbl> <chr>             <dbl> <chr>     <chr>
## 1  2013     1     1 517                 515 2         830
## 2  2013     1     1 533                 529 4         850
## 3  2013     1     1 542                 540 2         923
## 4  2013     1     1 544                 545 -1        1004
## 5  2013     1     1 554                 600 -6        812
## 6  2013     1     1 554                 558 -4        740
## # ... with 12 more variables: sched_arr_time <dbl>, arr_delay <chr>,
## #   carrier <chr>, flight <dbl>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <chr>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <chr>
```

```r
airlines <- read_excel("data/NYC_Flights_2013.xlsx", sheet = 2)
head(airlines)
```

```
## # A tibble: 6 x 2
##   carrier name
##   <chr>   <chr>
## 1 9E      Endeavor Air Inc.
## 2 AA      American Airlines Inc.
## 3 AS      Alaska Airlines Inc.
## 4 B6      JetBlue Airways
## 5 DL      Delta Air Lines Inc.
## 6 EV      ExpressJet Airlines Inc.
```

Chapter 6

Subsetting vectors and data frames in base R

First, we will create an example atomic vector to be used throughout the section. To do this, we will use the **sample** and **set.seed** functions. If you run the same code, you should have end up with the same values in your own vector.

```
set.seed(1001)
my_vec <- sample(1:20, 10, replace = TRUE)
my_vec
```

```
##  [1]  3 15 16  7 16 11  6 14  4 12
```

Subsetting vectors using

Using positive integer indices

A single positive integer index

Select the 2nd element in the vector

```
my_vec[2]
```

```
## [1] 15
```

A vector of positive integer indices

Select the 4th, 3rd and 7th elements of the vector (in that order).

```
my_vec[c(4,3,7)]
```

```
## [1]  7 16  6
```

A single negative index

We can omit an element of a vector by using a negative index. For example, to omit the 5th element of the vector we can run the following command

```
my_vec[-5]
```

```
## [1]  3 15 16  7 11  6 14  4 12
```

An array of negative indices

```
my_vec[c(-5, -9)]
```

```
## [1]  3 15 16  7 11  6 14 12
```

Alternatively, we could use the **-** outside the vector

```
my_vec[-c(5, 9)]
```

```
## [1]  3 15 16  7 11  6 14 12
```

Boolean Masking in vectors

A : a vector of logical values. The mask is ideally of the same length as the vector to be filtered.

```
mask <- my_vec > 8
mask
```

```
##  [1] FALSE  TRUE  TRUE FALSE  TRUE  TRUE FALSE  TRUE FALSE  TRUE
```

```
my_vec[mask]
```

```
## [1] 15 16 16 11 14 12
```

Locations in the data vector corresponding to locations of the mask that are *TRUE* are kept, while locations that correspond to values of *FALSE* in the mask are dropped.

The above steps could have been done in a single step as follows:

```
my_vec[my_vec > 8]
```

```
## [1] 15 16 16 11 14 12
```

Similarly, we could have filtered the data vector, to keep only those elements that are even numbers returned a remainder of 0 when divided by two

```
my_vec[my_vec%%2 == 0]
```

```
## [1] 16 16  6 14  4 12
```

Subsetting data frames

Using $ to extract a single column from a data frame

When working with data frames it is frequently useful to be able to reference a single column of the data frame. This can be done using the operator, **$**.

This operator can be used for

- reading or extracting a column
- creating a new column in a data frame
- overwriting the values of an existing column

```
library(notitia)
df <- areas
areas
```

```
## # A tibble: 7 x 2
##   country        area
##   <chr>         <dbl>
```

```
## 1 Russia        16376
## 2 China          9388
## 3 United States  9147
## 4 Brazil         8358
## 5 India          2973
## 6 Indonesia      1811
## 7 Nigeria         910
```

```
df$area
```

```
## [1] 16376  9388  9147  8358  2973  1811   910
```

We can create a new column in a data frame using the **$** on the left hand side of an assignment. A new column containing the areas of countries in millions of square miles can be added. We can do this by multiplying the areas by 0.386102

```
df$area_sqm <- df$area*0.386102
df
```

```
## # A tibble: 7 x 3
##   country        area area_sqm
##   <chr>         <dbl>    <dbl>
## 1 Russia        16376    6323.
## 2 China          9388    3625.
## 3 United States  9147    3532.
## 4 Brazil         8358    3227.
## 5 India          2973    1148.
## 6 Indonesia      1811     699.
## 7 Nigeria         910     351.
```

Overwriting a column

Lastly, we will give an example of using **$** to overwrite a column in a data frame. Currently the units of the *area* column are in millions of square kilometers. We can change the units so that the values in each column correspond to the land areas of the given countries in square kilometers. We do this by multiplying each element in the column by 1 million.

```
df$area <- df$area*1e6
df
```

```
## # A tibble: 7 x 3
##   country              area area_sqm
##   <chr>               <dbl>    <dbl>
## 1 Russia        16376000000    6323.
## 2 China          9388000000    3625.
## 3 United States  9147000000    3532.
## 4 Brazil         8358000000    3227.
## 5 India          2973000000    1148.
## 6 Indonesia      1811000000     699.
## 7 Nigeria         910000000     351.
```

```
1e6 - 1000000
```

```
## [1] 0
```

Using with data frames

In my experience, one typically

Chapter 7

Commands for exploring data

```
library(notitia)
```

dim the size of a data frame

```
dim(lara)
```

```
## [1] 561   8
```

str: the structure of a data frame

```
str(lara)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    561 obs. of  8 variables:
##  $ Runs      : int  11 44 5 23 5 45 0 54 18 45 ...
##  $ Inning    : Factor w/ 2 levels "1","2": 1 1 2 1 1 1 1 1 1 1 ...
##  $ Notout    : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ DNB       : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ Opp       : chr  "Pakistan" "Pakistan" "Pakistan" "England" ...
##  $ Ground    : chr  "Karachi" "Lahore" "Lahore" "Lord's" ...
##  $ Start Date: chr  "9-Nov-90" "6-Dec-90" "6-Dec-90" "27-May-91" ...
##  $ Match     : chr  "ODI # 639" "Test # 1158" "Test # 1158" "ODI # 678" ...
```

summary: summary statistics for a data frame

```
summary(lara)
```

```
##       Runs         Inning    Notout           DNB
##  Min.   :  0.00   1:430   Mode :logical   Mode :logical
##  1st Qu.:  9.00   2:131   FALSE:501       FALSE:543
##  Median : 29.00           TRUE :38        TRUE :18
##  Mean   : 42.91           NA's :22
##  3rd Qu.: 60.00
##  Max.   :400.00
##  NA's   :40
##      Opp               Ground           Start Date
##  Length:561         Length:561         Length:561
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Match
##  Length:561
##  Class :character
##  Mode  :character
##
##
##
##
```

head and tail:

```
head(lara)
```

```
## # A tibble: 6 x 8
##    Runs Inning Notout DNB   Opp      Ground  `Start Date` Match
##   <int> <fct>  <lgl>  <lgl> <chr>    <chr>   <chr>        <chr>
## 1    11 1      FALSE  FALSE Pakistan Karachi 9-Nov-90     ODI # 639
## 2    44 1      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 3     5 2      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 4    23 1      FALSE  FALSE England  Lord's  27-May-91    ODI # 678
## 5     5 1      FALSE  FALSE Pakistan Sharjah 17-Oct-91    ODI # 679
## 6    45 1      FALSE  FALSE India    Sharjah 19-Oct-91    ODI # 681
```

```
tail(lara)
```

```
## # A tibble: 6 x 8
##    Runs Inning Notout DNB   Opp          Ground      `Start Date` Match
##   <int> <fct>  <lgl>  <lgl> <chr>        <chr>       <chr>        <chr>
## 1    77 1      FALSE  FALSE Australia    North Sound 27-Mar-07    ODI # 25~
## 2    37 1      FALSE  FALSE New Zealand  North Sound 29-Mar-07    ODI # 25~
## 3     2 1      FALSE  FALSE Sri Lanka    Providence  1-Apr-07     ODI # 25~
## 4    21 1      FALSE  FALSE South Africa St George's 10-Apr-07    ODI # 25~
## 5    33 1      FALSE  FALSE Bangladesh   Bridgetown  19-Apr-07    ODI # 25~
## 6    18 1      FALSE  FALSE England      Bridgetown  21-Apr-07    ODI # 25~
```

```
head(lara, 8)
```

```
## # A tibble: 8 x 8
##    Runs Inning Notout DNB   Opp      Ground  `Start Date` Match
##   <int> <fct>  <lgl>  <lgl> <chr>    <chr>   <chr>        <chr>
## 1    11 1      FALSE  FALSE Pakistan Karachi 9-Nov-90     ODI # 639
## 2    44 1      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 3     5 2      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 4    23 1      FALSE  FALSE England  Lord's  27-May-91    ODI # 678
## 5     5 1      FALSE  FALSE Pakistan Sharjah 17-Oct-91    ODI # 679
## 6    45 1      FALSE  FALSE India    Sharjah 19-Oct-91    ODI # 681
## 7     0 1      FALSE  FALSE Pakistan Sharjah 21-Oct-91    ODI # 682
## 8    54 1      FALSE  FALSE Pakistan Karachi 20-Nov-91    ODI # 689
```

table: getting counts of variable values

```
table(chi_emps$Department)
```

```
##
##                ADMIN HEARNG               ANIMAL CONTRL
##                          36                          78
##                    AVIATION           BOARD OF ELECTION
##                        1670                         108
##             BOARD OF ETHICS               BUDGET & MGMT
##                           8                          43
##                   BUILDINGS            BUSINESS AFFAIRS
##                         269                         171
##                  CITY CLERK                CITY COUNCIL
##                          94                         382
```

```
##                      COPA            CULTURAL AFFAIRS
##                       124                          75
##              DISABILITIES                       DoIT
##                        27                          99
##          FAMILY & SUPPORT                     FINANCE
##                       632                         575
##                      FIRE FLEET & FACILITY MANAGEMENT
##                      4633                         971
##                    HEALTH                     HOUSING
##                       474                          59
##           HUMAN RELATIONS            HUMAN RESOURCES
##                        18                          80
##             INSPECTOR GEN                         LAW
##                        83                         394
##         LICENSE APPL COMM             MAYOR'S OFFICE
##                         1                          76
##                      OEMC   PLANNING AND DEVELOPMENT
##                      1950                         154
##                    POLICE               POLICE BOARD
##                     14083                           2
##               PROCUREMENT             PUBLIC LIBRARY
##                        87                         960
##             STREETS & SAN                 TRANSPORTN
##                      2206                        1146
##                TREASURER                WATER MGMNT
##                        24                        1900
```

```
table(chi_emps$Department, chi_emps$FullPart)
```

```
##
##                                F    P
##   ADMIN HEARNG                36    0
##   ANIMAL CONTRL               63   15
##   AVIATION                  1605   65
##   BOARD OF ELECTION          108    0
##   BOARD OF ETHICS              8    0
##   BUDGET & MGMT               43    0
##   BUILDINGS                  269    0
##   BUSINESS AFFAIRS           164    7
##   CITY CLERK                  94    0
##   CITY COUNCIL               318   64
##   COPA                       124    0
##   CULTURAL AFFAIRS            75    0
##   DISABILITIES                27    0
##   DoIT                        99    0
```

```
##   FAMILY & SUPPORT              310   322
##   FINANCE                       571     4
##   FIRE                         4633     0
##   FLEET & FACILITY MANAGEMENT   971     0
##   HEALTH                        474     0
##   HOUSING                        59     0
##   HUMAN RELATIONS                18     0
##   HUMAN RESOURCES                80     0
##   INSPECTOR GEN                  83     0
##   LAW                           392     2
##   LICENSE APPL COMM               1     0
##   MAYOR'S OFFICE                 76     0
##   OEMC                          847  1103
##   PLANNING AND DEVELOPMENT      151     3
##   POLICE                      14060    23
##   POLICE BOARD                    2     0
##   PROCUREMENT                    84     3
##   PUBLIC LIBRARY                710   250
##   STREETS & SAN                2048   158
##   TRANSPORTN                   1146     0
##   TREASURER                      23     1
##   WATER MGMNT                  1899     1
```

```
table(chi_emps$FullPart, chi_emps$Department)
```

```
##
##     ADMIN HEARNG ANIMAL CONTRL AVIATION BOARD OF ELECTION BOARD OF ETHICS
##   F           36            63     1605               108               8
##   P            0            15       65                 0               0
##
##     BUDGET & MGMT BUILDINGS BUSINESS AFFAIRS CITY CLERK CITY COUNCIL  COPA
##   F            43       269              164         94          318   124
##   P             0         0                7          0           64     0
##
##     CULTURAL AFFAIRS DISABILITIES  DoIT FAMILY & SUPPORT FINANCE  FIRE
##   F               75           27    99              310     571  4633
##   P                0            0     0              322       4     0
##
##     FLEET & FACILITY MANAGEMENT HEALTH HOUSING HUMAN RELATIONS
##   F                         971    474      59              18
##   P                           0      0       0               0
##
##     HUMAN RESOURCES INSPECTOR GEN   LAW LICENSE APPL COMM MAYOR'S OFFICE
##   F              80            83   392                 1             76
##   P               0             0     2                 0              0
```

```
##
##      OEMC PLANNING AND DEVELOPMENT POLICE POLICE BOARD PROCUREMENT
##   F   847                      151  14060            2          84
##   P  1103                        3     23            0           3
##
##     PUBLIC LIBRARY STREETS & SAN TRANSPORTN TREASURER WATER MGMNT
##   F            710          2048       1146        23        1899
##   P            250           158          0         1           1
```

Chapter 8

Simple Plotting in R

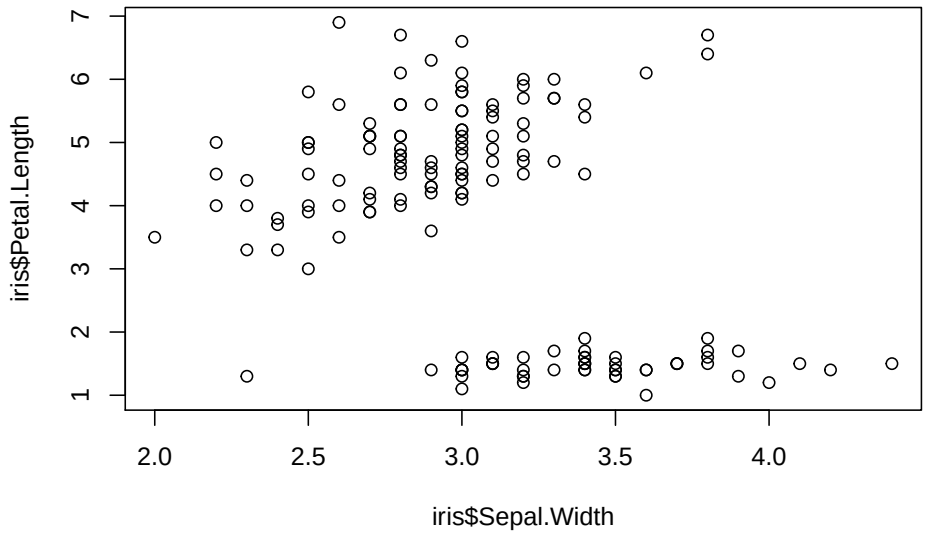
Plotting commands in base R

plot

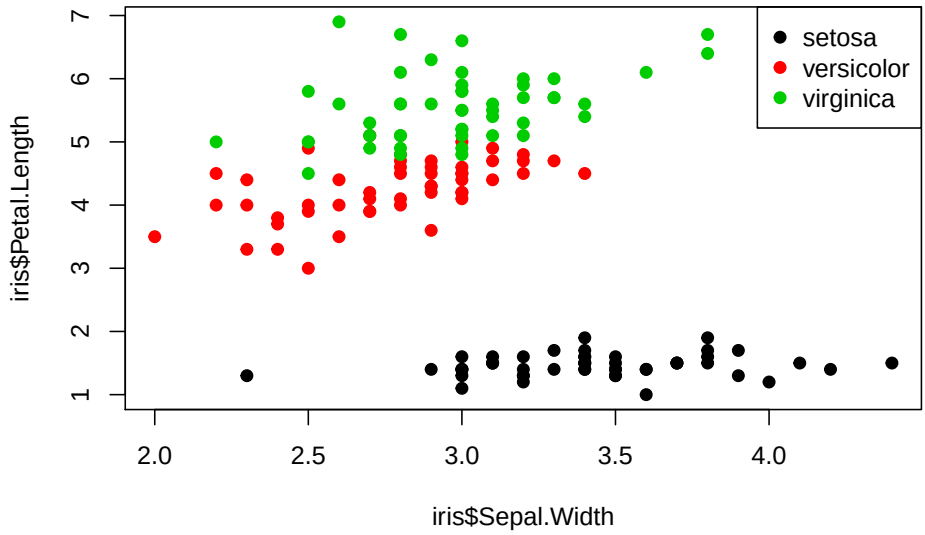
```
head(iris)
```

```
##   Sepal.Length 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
```

```
plot(iris$Sepal.Width, iris$Petal.Length)
```

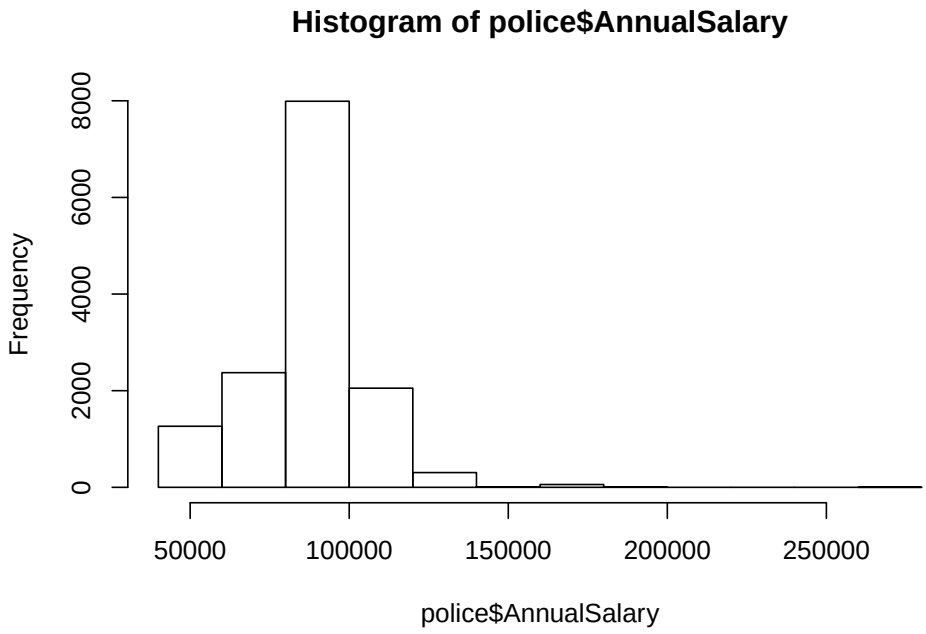



```
plot(iris$Sepal.Width, iris$Petal.Length, col = iris$Species, pch = 19)
legend("topright",legend=levels(iris$Species), col = 1:3, pch = 19)
```



hist

```
library(notitia)
police <- chi_emps[chi_emps$Department == "POLICE", ]
hist(police$AnnualSalary)
```



barplot

boxplot

Chapter 9

Plotting with qplot

```
library(ggplot2)
```

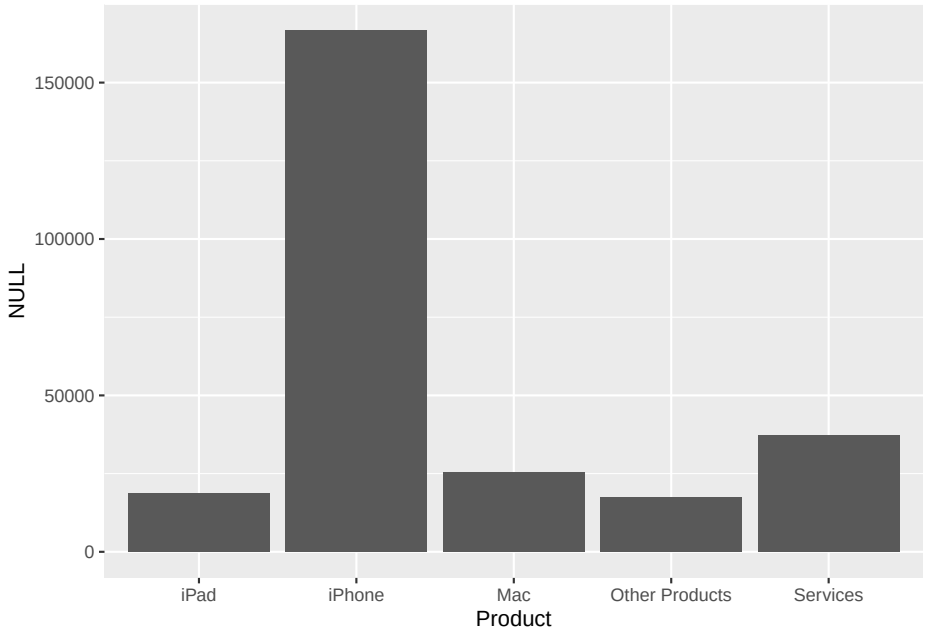
Quantities or Proportions

```
library(dplyr)
apple_2018 <- filter(apple, Year == 2018) %>%
  group_by(Product) %>% summarise(Revenue = sum(Revenue))
apple_2018
```

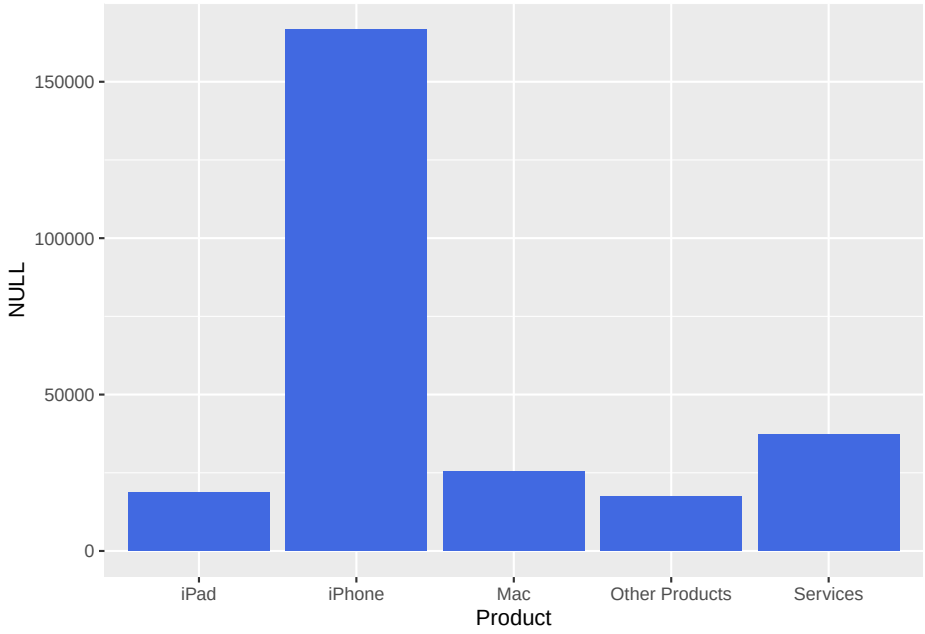
```
## # A tibble: 5 x 2
##   Product        Revenue
##   <chr>            <dbl>
## 1 iPad             18805
## 2 iPhone          166699
## 3 Mac              25484
## 4 Other Products   17417
## 5 Services         37190
```

Bar Charts

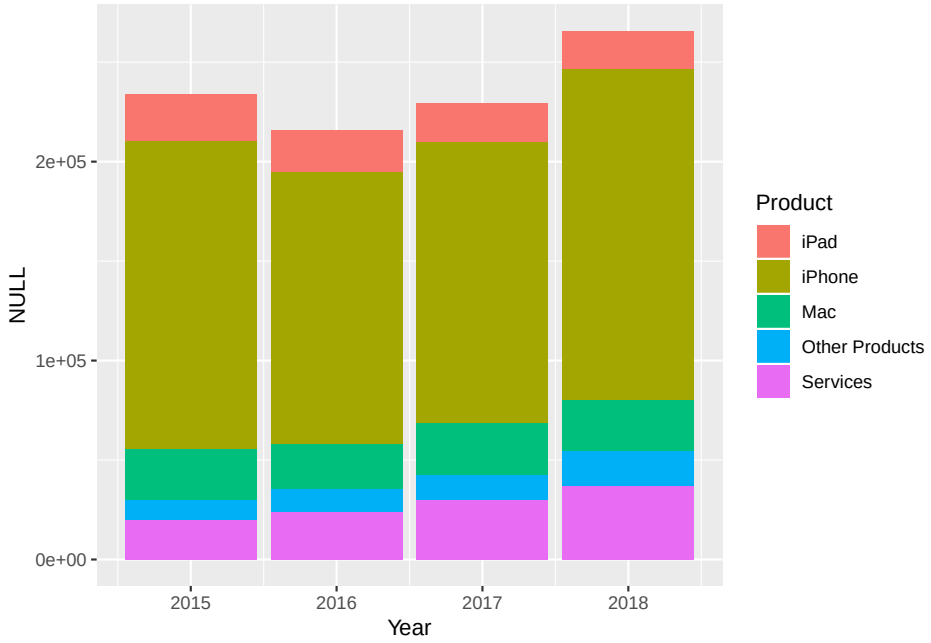
```
qplot(x = Product, data = apple_2018, geom = "bar", weight = Revenue)
```



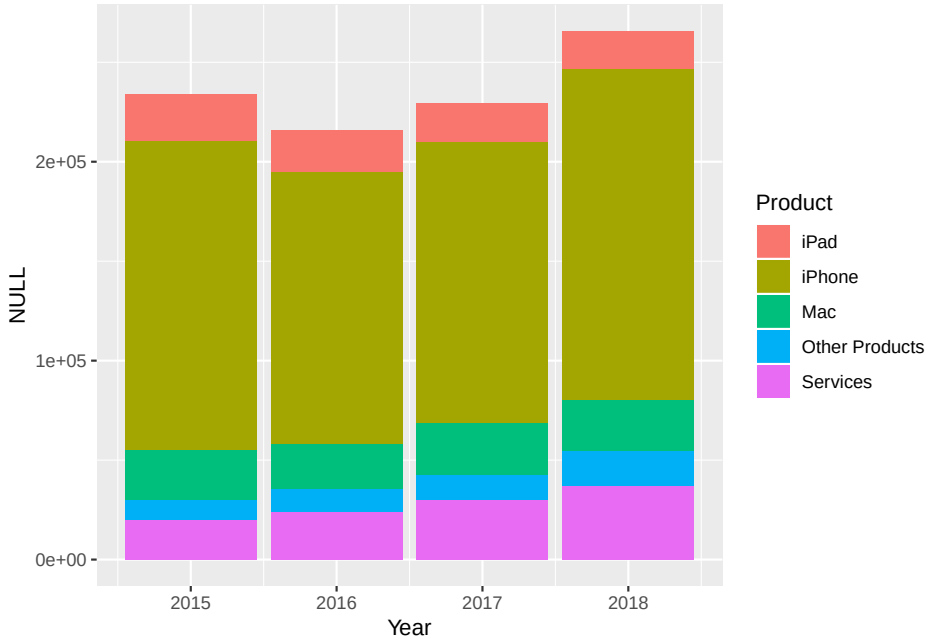

```
qplot(x = Product, data = apple_2018, geom = "bar", weight = Revenue, fill = I("royalb
```



```
qplot(x = Year, data = apple, geom = "bar", weight = Revenue, fill = Product)
```




```
qplot(x = Year, data = apple, geom = "bar", weight = Revenue, fill = Product)
```

Distributions

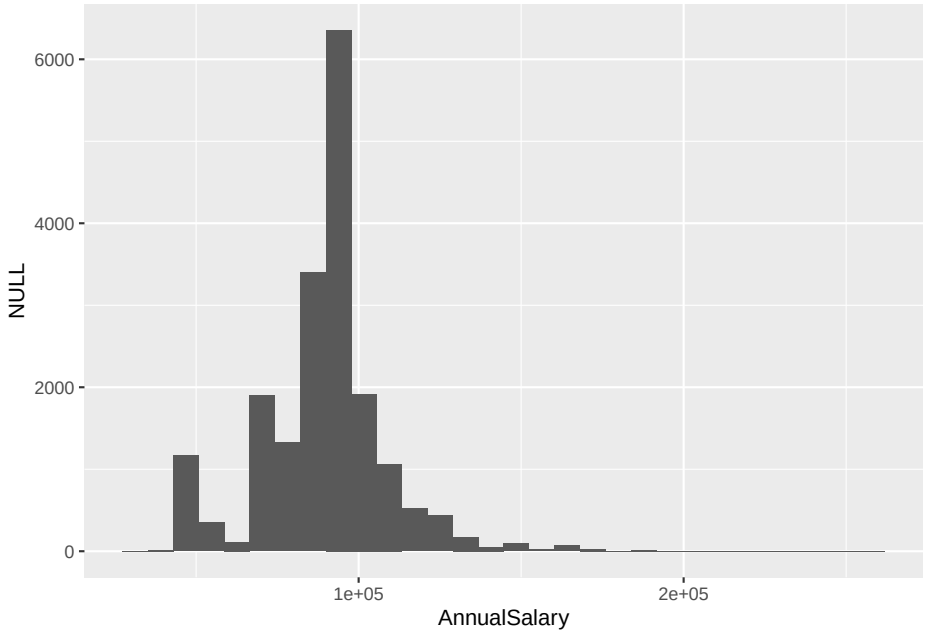
Geometries: - *histogram* - *boxplot* - *density*

```
library(notitia)
large_depts <- chi_emps[chi_emps$Department %in% c("POLICE", "FIRE", "STREETS & SAN"),
table(large_depts$Department)
```

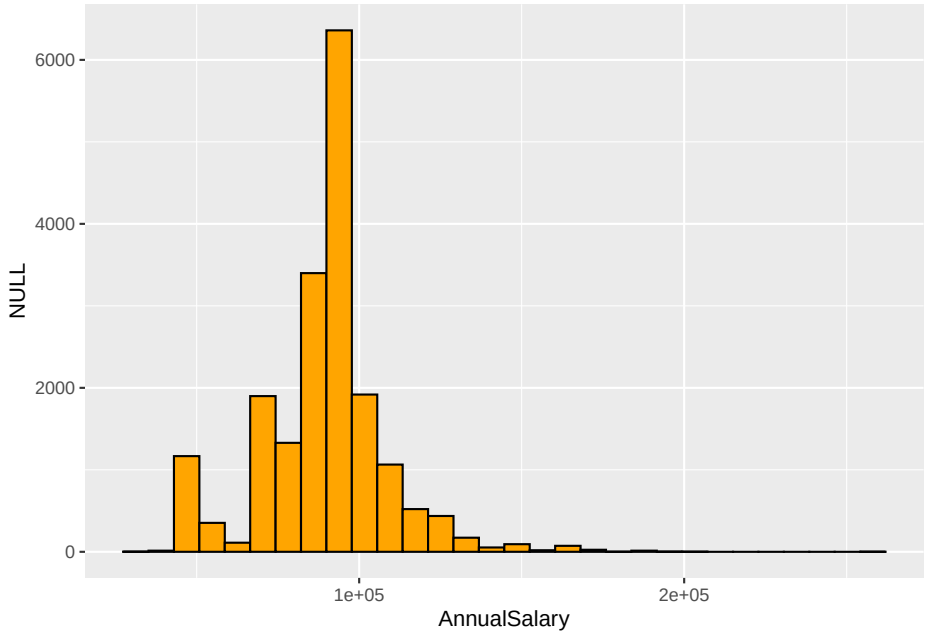
```
##
##          FIRE        POLICE STREETS & SAN
##          4633         14083          2206
```

Histograms

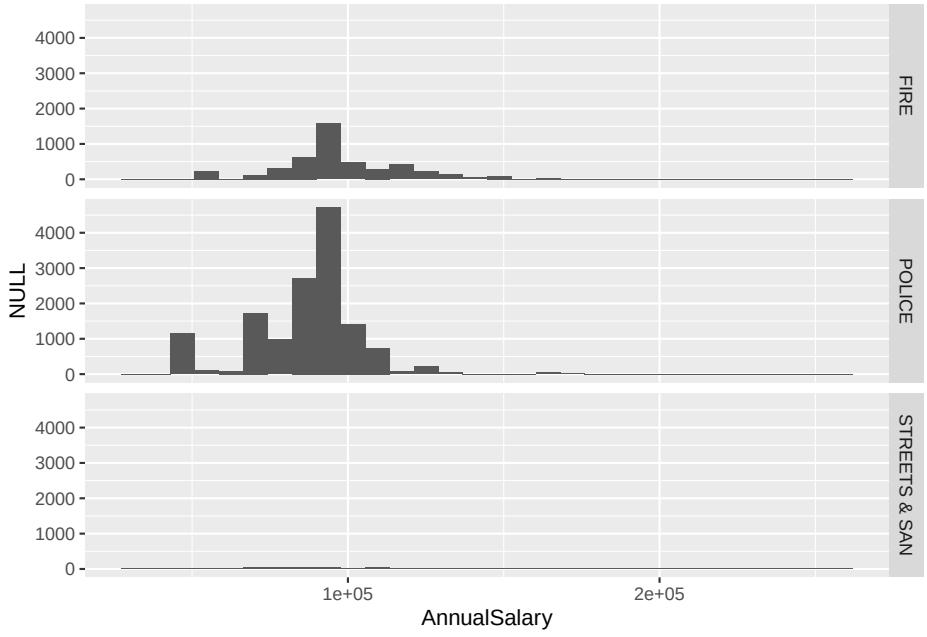
```
qplot(x = AnnualSalary, data = large_depts, geom = "histogram")
```



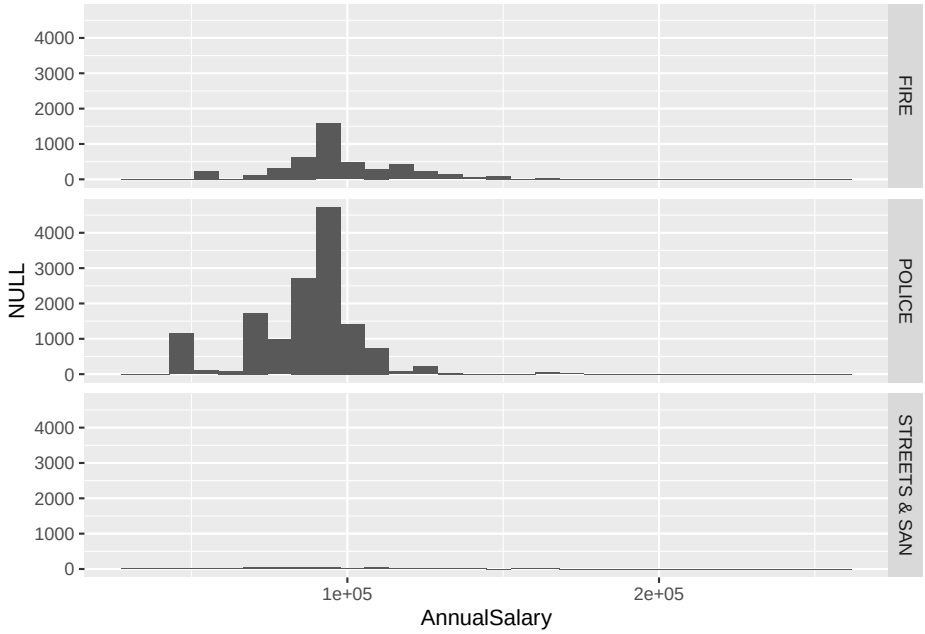

```
qplot(x = AnnualSalary, data = large_depts, geom = "histogram", fill = I("orange"), col
```




```
library(ggplot2)
qplot(x = AnnualSalary, data = large_depts, geom = "histogram", facets = Department~.)
```

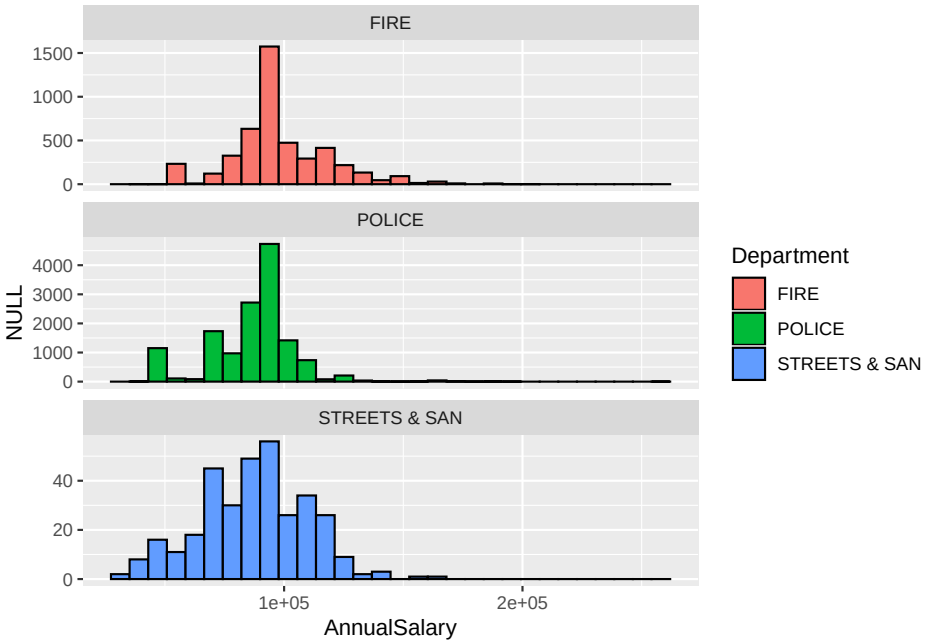


```
qplot(x = AnnualSalary, data = large_depts, geom = "histogram", facets = Department~.,
```



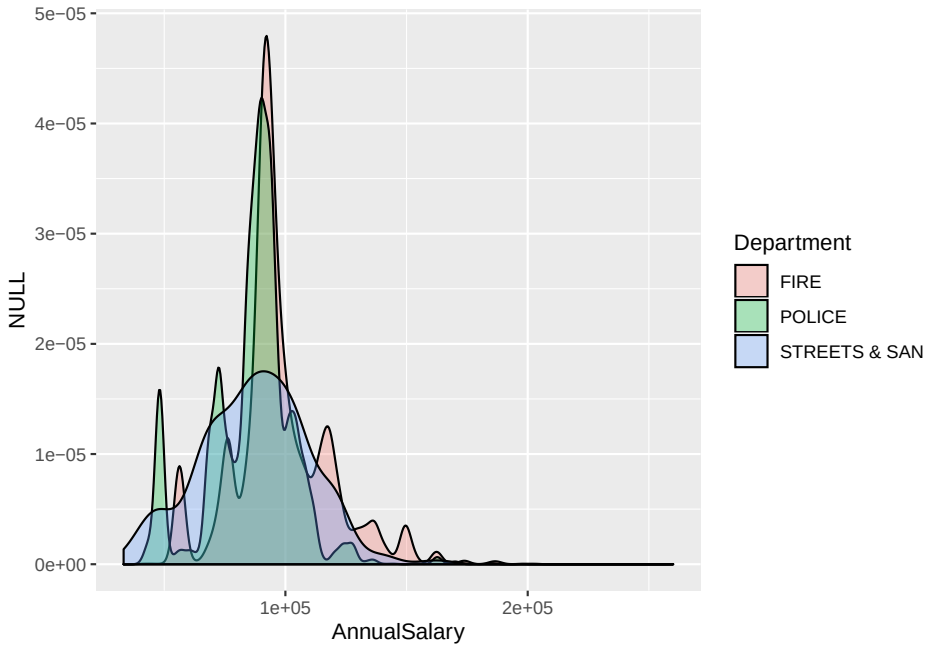

```
qplot(x = AnnualSalary, data = large_depts, geom = "histogram", fill = Department) +
  facet_wrap(Department~., scales = "free_y", ncol = 1)
```


```
qplot(x = AnnualSalary, data = large_depts, geom = "histogram", fill = Department, colour = I("bl
  facet_wrap(Department~., scales = "free_y", ncol = 1)
```

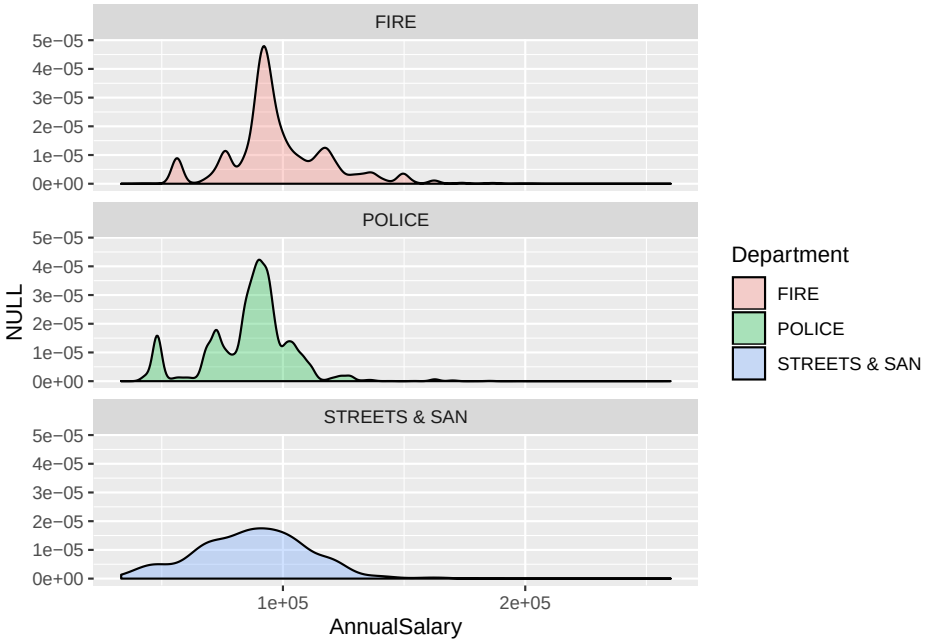


Density plots

```
qplot(x = AnnualSalary, data = large_depts, geom = "density", fill = Department, alpha
```

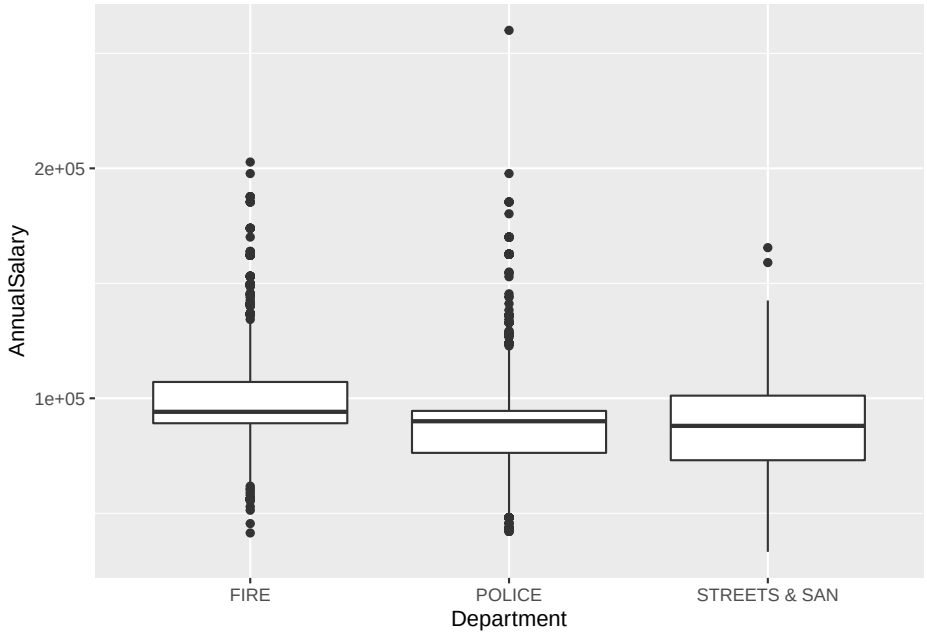



```
qplot(x = AnnualSalary, data = large_depts, geom = "density", fill = Department, alpha
  facet_wrap(Department~., ncol = 1)
```



Box plots

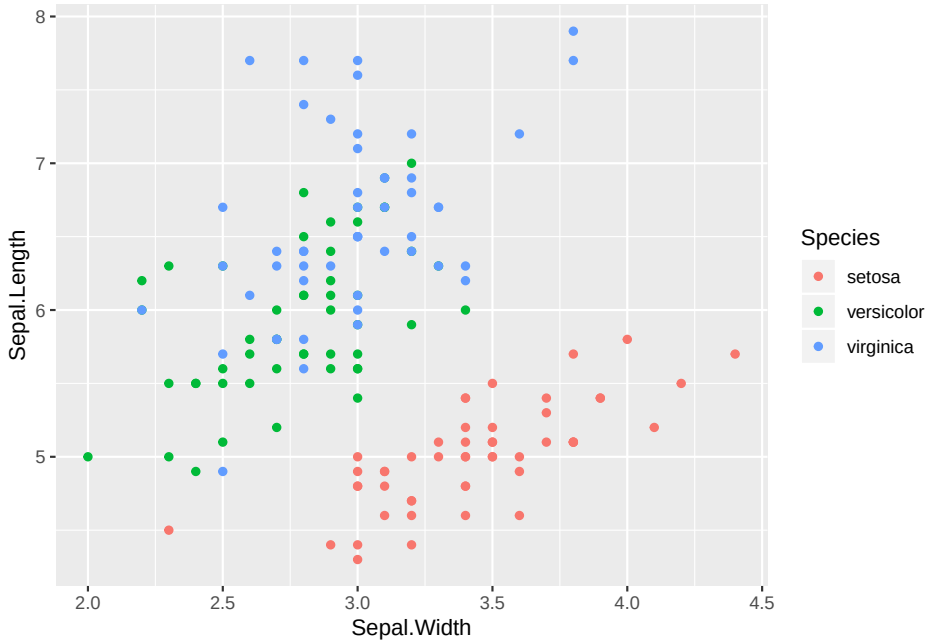
```
qplot(x = Department, y = AnnualSalary, data = large_depts, geom = "boxplot")
```



x-y relationships

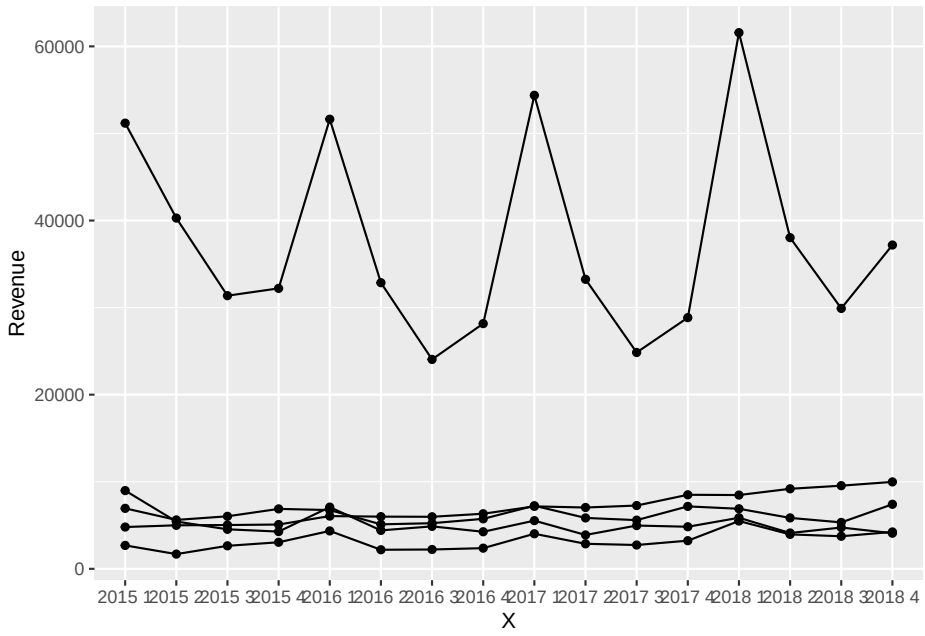
```
qplot(x = Sepal.Width, y = Sepal.Length, data = iris, geom = "point")
```


```
qplot(x = Sepal.Width, y = Sepal.Length, data = iris, colour = Species, geom = "point")
```

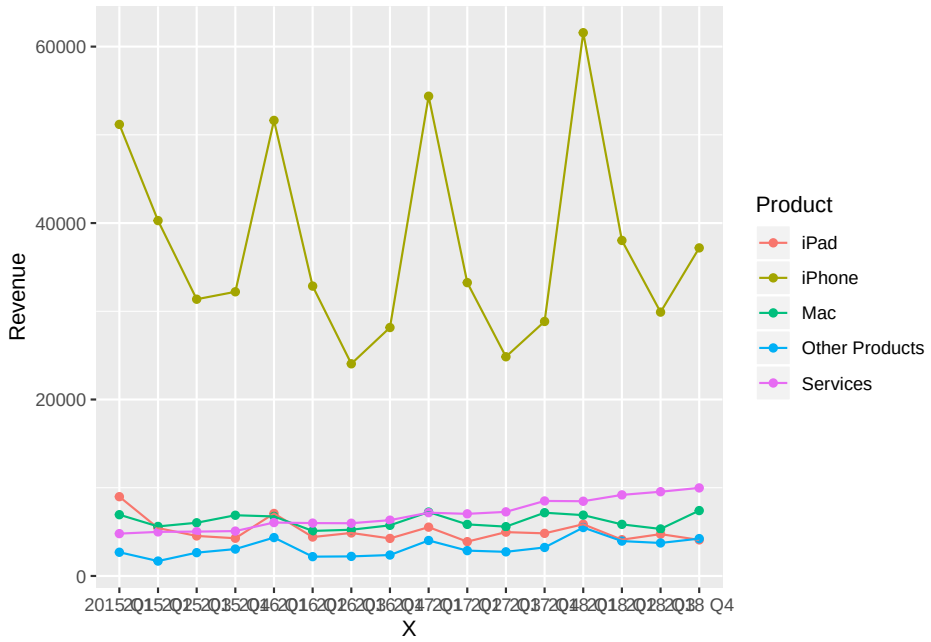


Time Series

```
apple$X = paste(apple$Year, apple$Quarter)
qplot(x = X, y = Revenue, data = apple, group = Product, geom = c("point", "line"))
```



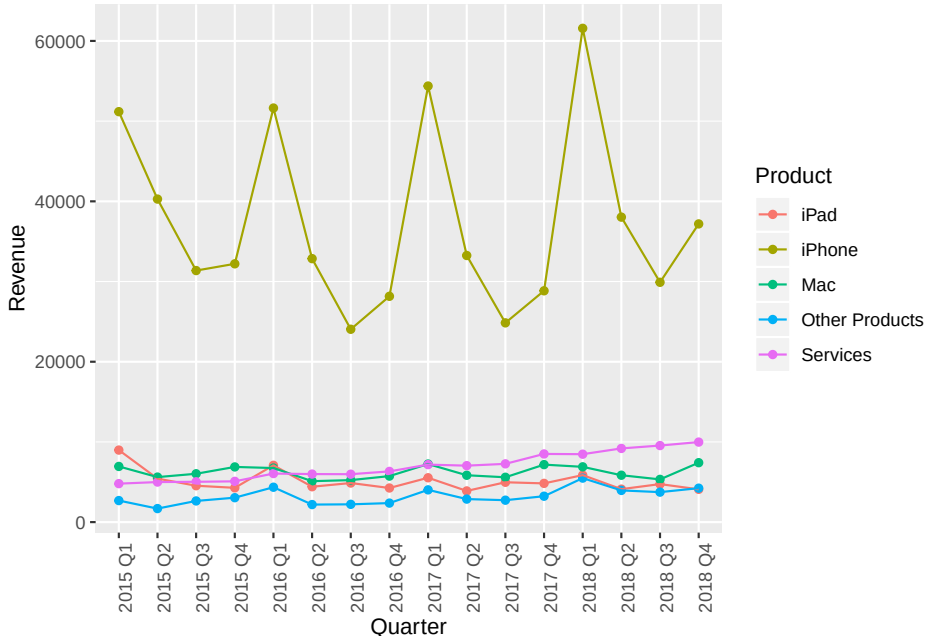

```
apple$X = paste(apple$Year, apple$Quarter, sep = " Q")
qplot(x = X, y = Revenue, data = apple, group = Product, colour = Product, geom = c("po
```




```
df <- apple
df$Quarter = paste(apple$Year, apple$Quarter, sep = " Q")
head(df)
```

```
## # A tibble: 6 x 6
##    Year Quarter Product        Units Revenue X
##   <int> <chr>   <chr>          <dbl>   <dbl> <chr>
## 1  2015 2015 Q1 iPad           21419    8985 2015 Q1
## 2  2015 2015 Q1 iPhone         74468   51182 2015 Q1
## 3  2015 2015 Q1 Mac             5519    6944 2015 Q1
## 4  2015 2015 Q1 Other Products    NA    2689 2015 Q1
## 5  2015 2015 Q1 Services          NA    4799 2015 Q1
## 6  2015 2015 Q2 iPad           12623    5428 2015 Q2
```

```
qplot(x = Quarter, y = Revenue, data = df, group = Product, colour = Product, geom = c("point", "
```


60000
40000
20000
0
Revenue
2015 Q1
2015 Q2
2015 Q3
2015 Q4
2016 Q1
2016 Q2
2016 Q3
2016 Q4
2017 Q1
2017 Q2
2017 Q3
2017 Q4
2018 Q1
2018 Q2
2018 Q3
2018 Q4
Quarter
Product
iPad
iPhone
Mac
Other Products
Services

Chapter 10

Intro to ggplot2

```
library(ggplot2)
```

```
library(scales)
```

- ggplot
  - data
  - aesthetic: a mapping
- geometry

aesthetics

- x: 1st variable
- y: 2nd variable

- group: variable
- colour/color

```
library(dplyr)
library(notitia)
```

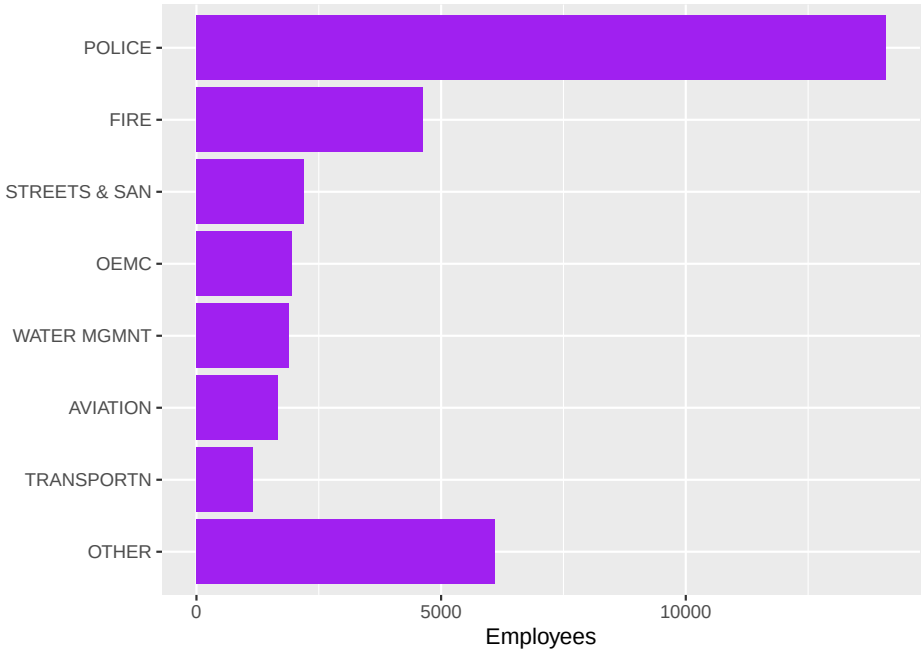
ggplot2 Geometries

Geometries for displaying quantities (or proportions)

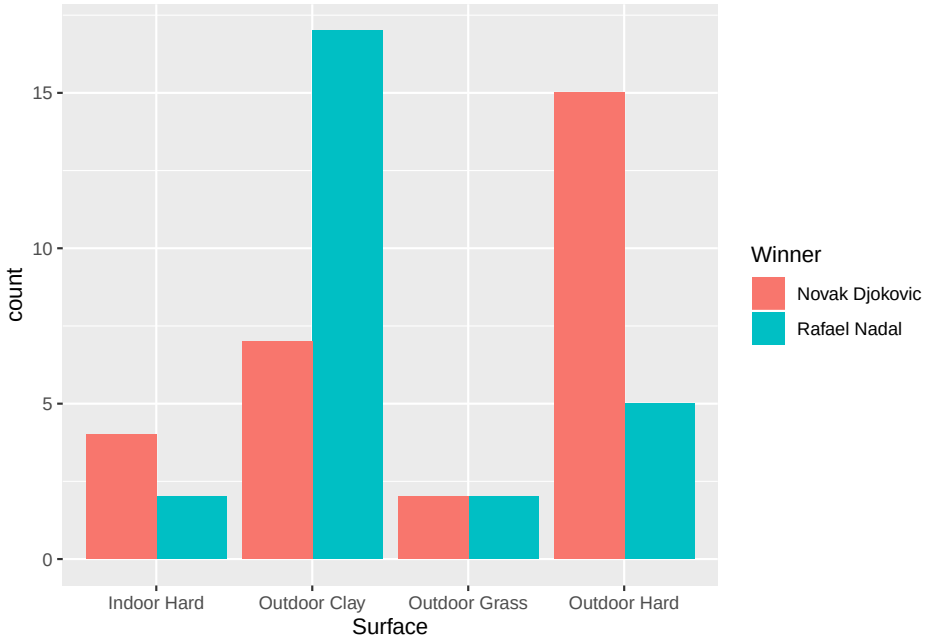
geom_bar

```
chicago <- chi_emps
dept_counts <- table(chicago$Department)
dept_counts <- sort(dept_counts[dept_counts > 1000])
dept_names <- names(dept_counts)
chicago$Dept <- ifelse(chicago$Department %in% dept_names, chicago$Department, "OTHER")
chicago$Dept <- factor(chicago$Dept, levels = c("OTHER", dept_names))


ggplot(data = chicago, aes(x = Dept)) +
  geom_bar(fill = "purple") +
  coord_flip() + xlab("") + ylab("Employees")
```

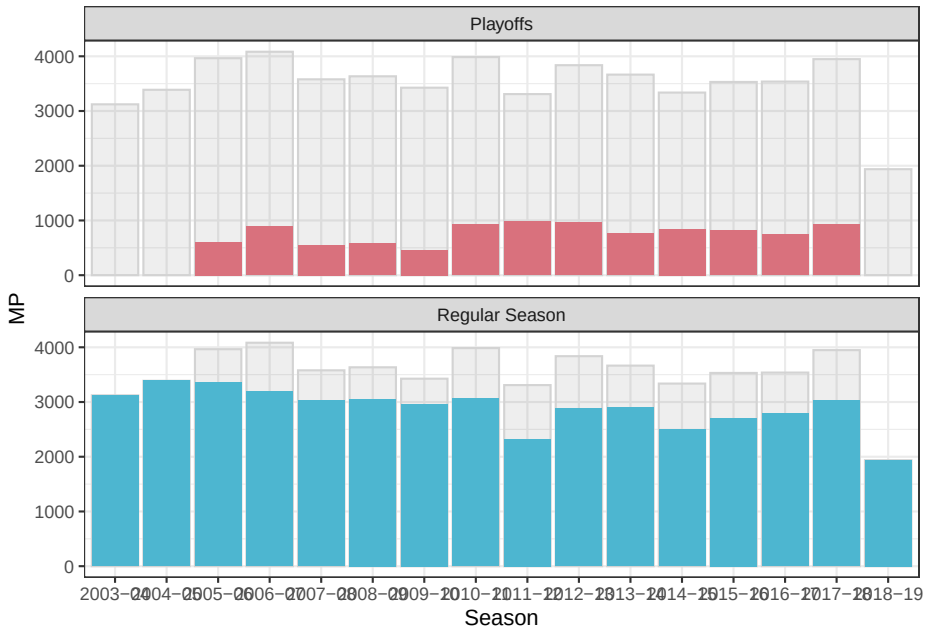



```
ggplot(rafa_novak, aes(x = Surface, fill = Winner)) + geom_bar(position = "dodge")
```




```
lbj_reg <- select(lebron, Season, MP) %>% mutate(RegPlayoffs = "Regular Season")
lbj_playoffs <- select(lebron_playoffs, Season, MP) %>% mutate(RegPlayoffs = "Playoffs")
lbj_mp <- bind_rows(lbj_reg, lbj_playoffs) %>% filter(Season != "Career")
lbj_mp_totals <- group_by(lbj_mp, Season) %>% summarise(MP = sum(MP))


ggplot(data = lbj_mp, aes(x = Season, y = MP)) +
  geom_bar(data = lbj_mp_totals, colour = "lightgrey", stat = "identity", alpha = .1) +
  geom_bar(stat = "identity", mapping = aes(fill = RegPlayoffs)) +
  guides(fill = FALSE) +
  facet_wrap(~ RegPlayoffs, ncol = 1) +
  scale_fill_manual(values = c("#D9717D", "#4DB6D0")) +
  theme_bw()
```



https://michaeltoth.me/a-detailed-guide-to-ggplot-colors.html

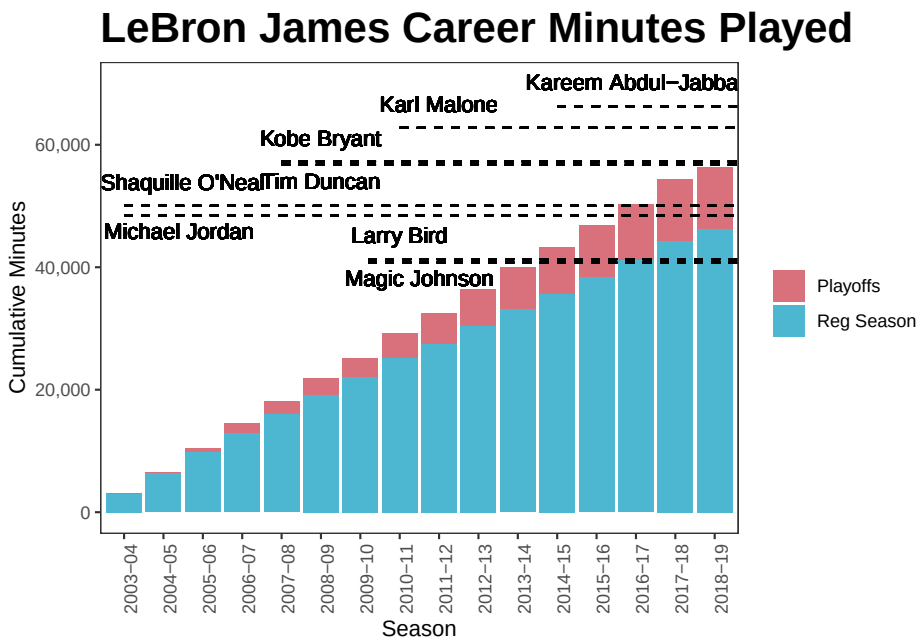
```
library(tidyr)
lbj_reg_min <- select(lebron, Season, MP) %>% rename(MPR = MP)
lbj_playoffs_min <- select(lebron_playoffs, Season, MP) %>% rename(MPP = MP)


lbj_all_min <- full_join(lbj_reg_min, lbj_playoffs_min, by = "Season") %>%
  filter(Season != "Career") %>%
  mutate(MPR = if_else(is.na(MPR), 0, MPR),
         MPP = if_else(is.na(MPP), 0, MPP),
         Playoffs = cumsum(MPP),
         `Reg Season` = cumsum(MPR)) %>%
  gather(key = RegPlayoffs, value = MP, Playoffs:`Reg Season`)
```

```
minplot <- ggplot(data = lbj_all_min, aes(x = Season, y = MP, fill = RegPlayoffs)) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = c("#D9717D", "#4DB6D0")) +
  theme_bw() +
  geom_segment(x = 12, xend = 17, y=66297, yend = 66297, linetype="dashed") + geom_text
  geom_segment(x = 8, xend = 17, y = 62759, yend = 62759, linetype="dashed") + geom_te
  geom_segment(x = 5, xend = 17, y = 57278, yend = 57278, linetype="dashed") + geom_te
  geom_segment(x = 5, xend = 17, y = 56738, yend = 56738, linetype="dashed") + geom_te
  geom_segment(x = 1, xend = 17, y = 50016, yend = 50016, linetype="dashed") + geom_te
  geom_segment(x = 1, xend = 17, y = 48485, yend = 48485, linetype="dashed") + geom_te
```

```
  geom_segment(x = 7.2, xend = 17, y = 41329, yend = 41329, linetype="dashed") + geom_text(aes(8,
   geom_segment(x = 7.2, xend = 17, y = 40783, yend = 40783, linetype="dashed") + geom_text(aes(8
  scale_y_continuous(label=comma, limits = c(0,70000)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.text.x = element_text(angle = 90),
        plot.title = element_text(size = 21, face = "bold"),
        legend.title = element_blank()
        )+
  ylab("Cumulative Minutes") +
  ggtitle("LeBron James Career Minutes Played")

minplot
```

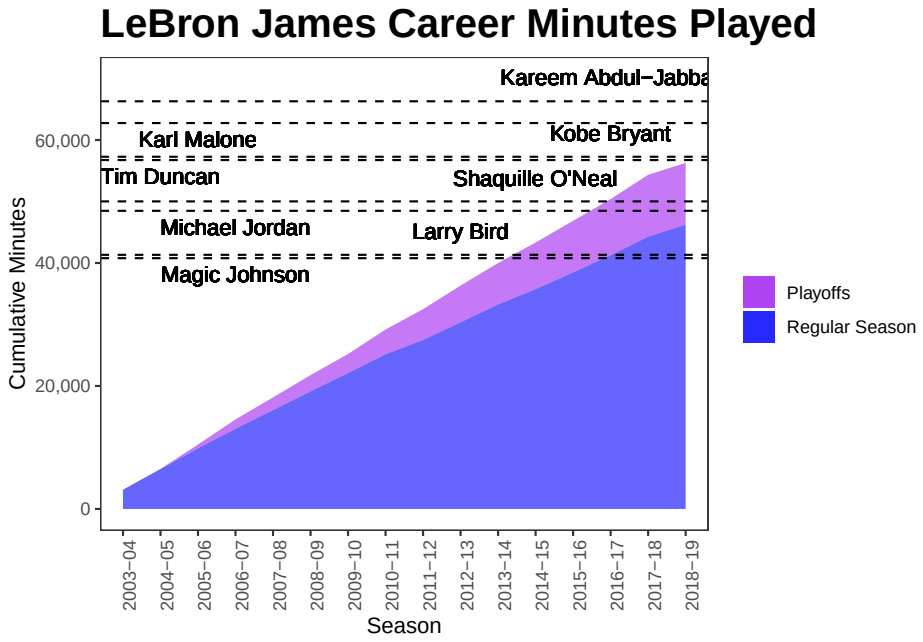



```
lbj<- full_join(lbj_reg_min, lbj_playoffs_min, by = "Season") %>%
  filter(Season != "Career") %>%
  mutate(MPR = if_else(is.na(MPR), 0, MPR),
         MPP = if_else(is.na(MPP), 0, MPP),
         Playoffs = cumsum(MPP),
         `Reg Season` = cumsum(MPR))


minplot2 <- ggplot(data = lbj, aes(x = Season)) +
  geom_ribbon(aes(ymin = 0, ymax = `Reg Season`, fill = "Regular Season"),
              group = 1, alpha = 0.6) +
```

```
  geom_ribbon(aes(ymin = `Reg Season`, ymax = `Reg Season` + Playoffs, fill = "Playoffs
              group = 1, alpha = 0.6) +
  theme_bw() +
  scale_fill_manual(values = c("purple", "blue"), name = "") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        axis.text.x = element_text(angle = 90),
        plot.title = element_text(size = 20, face = "bold")) +
  ylab("Cumulative Minutes") + scale_y_continuous(label=comma, limits = c(0,70000)) +
  geom_hline(yintercept = 66297, linetype="dashed") +
  geom_text(aes(14,66297,label = "Kareem Abdul-Jabbar", vjust = -1)) +
  geom_hline(yintercept = 62759, linetype="dashed") +
  geom_text(aes(3,62759,label = "Karl Malone", vjust = 1.5)) +
  geom_hline(yintercept = 57278, linetype="dashed") +
  geom_text(aes(14,57278,label = "Kobe Bryant", vjust = -1)) +
  geom_hline(yintercept = 56738, linetype="dashed") +
  geom_text(aes(2,56738,label = "Tim Duncan", vjust = 1.5)) +
  geom_hline(yintercept = 50016, linetype="dashed") +
  geom_text(aes(12,50016,label = "Shaquille O'Neal", vjust = -1)) +
  geom_hline(yintercept = 48485, linetype="dashed") +
  geom_text(aes(4,48485,label = "Michael Jordan", vjust = 1.5)) +
  geom_hline(yintercept = 41329, linetype="dashed") +
  geom_text(aes(10,41329,label = "Larry Bird", vjust = -1)) +
  geom_hline(yintercept = 40783, linetype="dashed") +
  geom_text(aes(4,40783,label = "Magic Johnson", vjust = 1.5)) +
  ggtitle("LeBron James Career Minutes Played")

minplot2
```

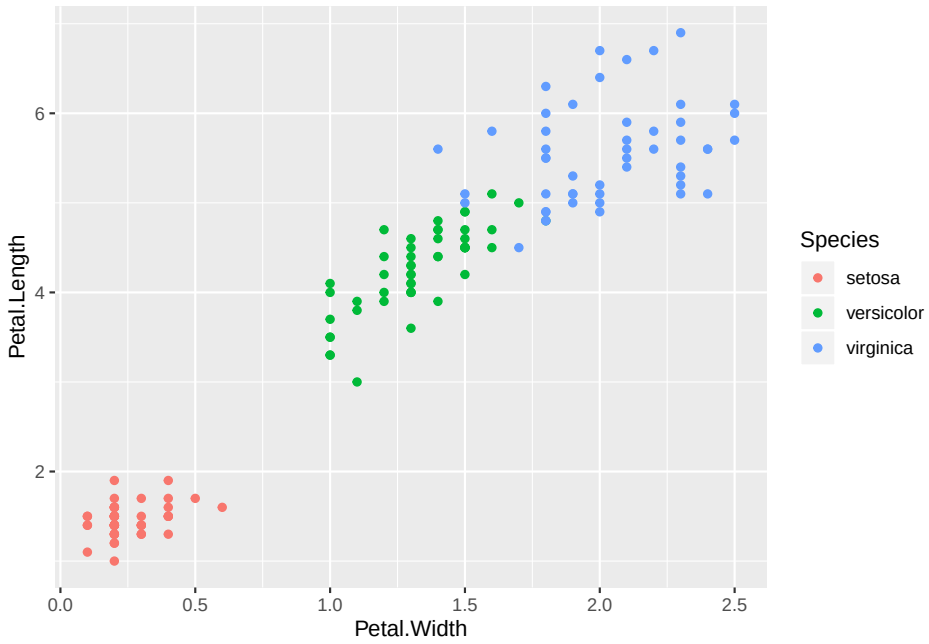



```
## Saving 6.5 x 4.5 in image
## Saving 6.5 x 4.5 in image
```

Geometries for showing x-y relationships

geom_point: for scatter plots

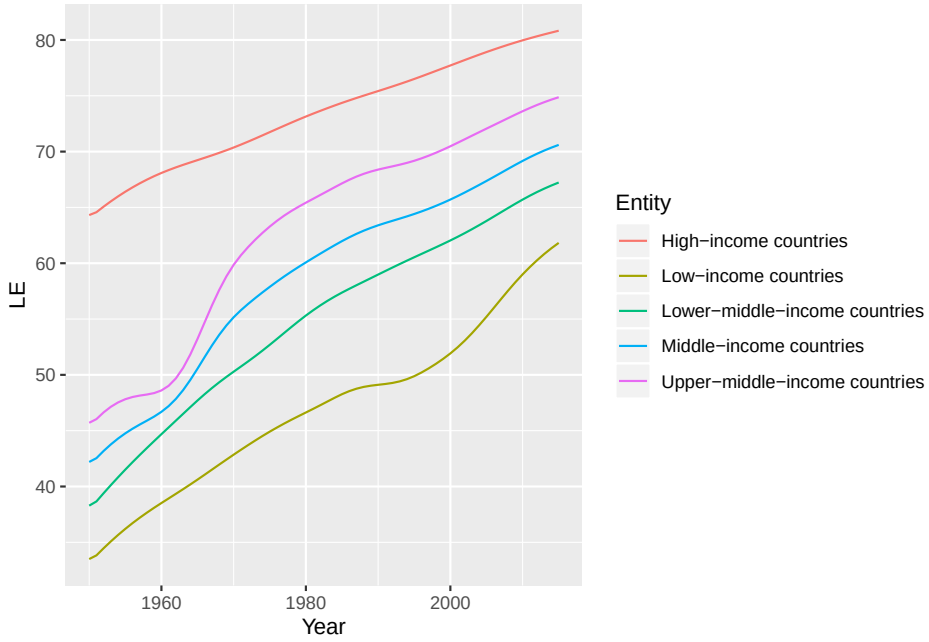
```
ggplot(data = iris, aes(x = Petal.Width, y = Petal.Length, colour = Species)) +
  geom_point()
```



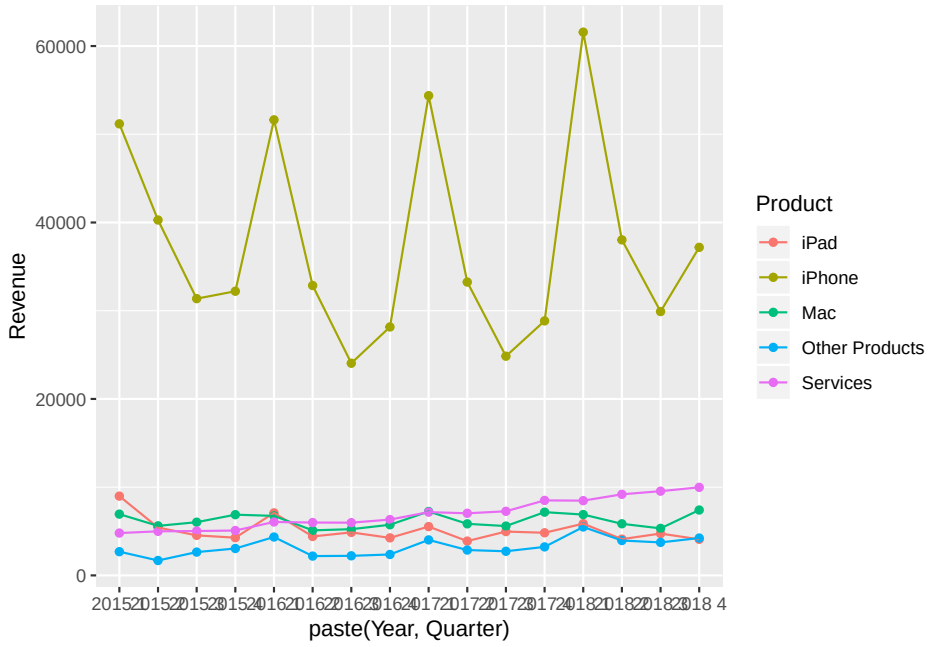
geom_line: for line charts

```
group_ex <- life_ex %>% filter(grepl("income countries", Entity))

ggplot(data = group_ex, aes(x = Year, y = LE, group = Entity, colour = Entity)) +
  geom_line()
```



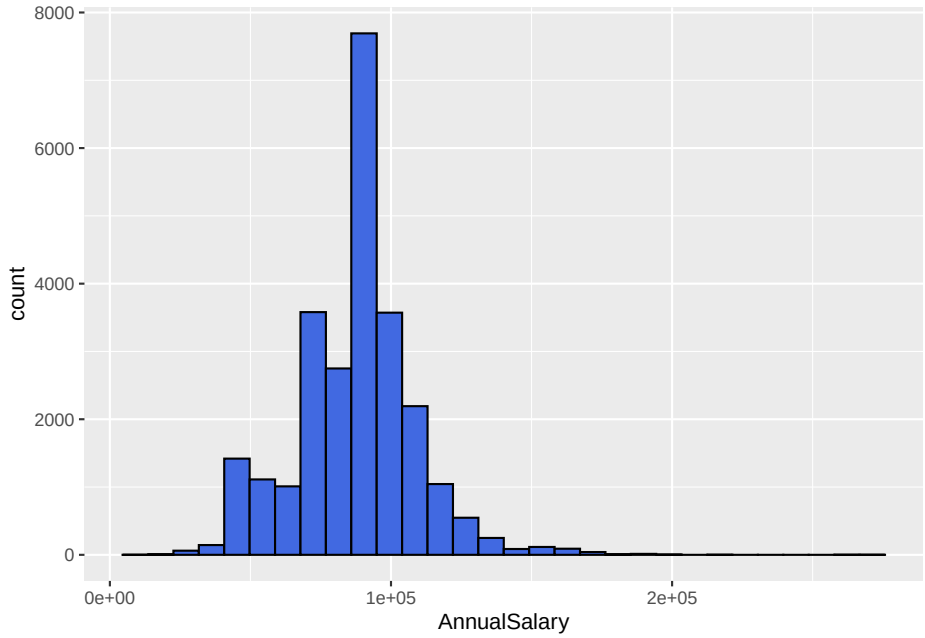

```
ggplot(data = apple, aes(x = paste(Year, Quarter), y = Revenue, group = Product, colour = Product
  geom_point() +
  geom_line()
```



Geometries for showing distributions

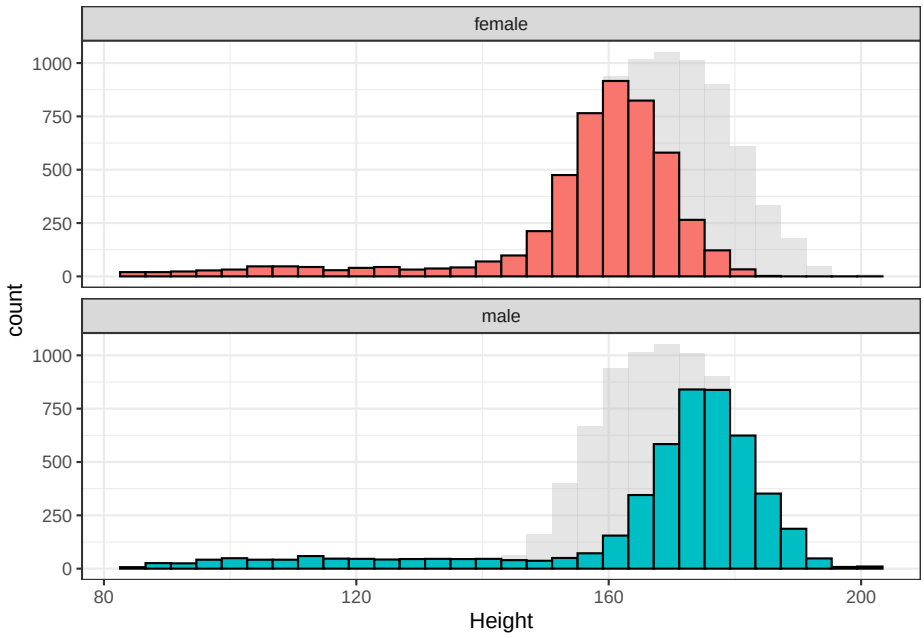
geom_histogram: for histograms

```
ggplot(data = chi_emps, aes(x = AnnualSalary)) +
  geom_histogram(fill = I("royalblue"), colour = I("black"))
```



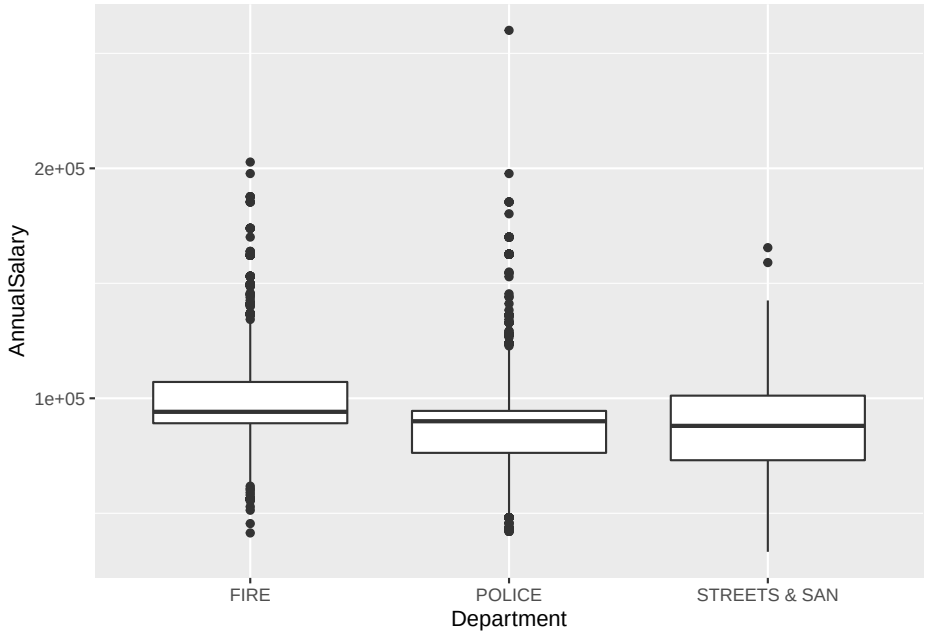

```
library(NHANES)
NHANES_bg <- select(NHANES, -Gender) %>%
  filter(Age >= 18)


ggplot(data = NHANES, aes(x = Height)) +
  geom_histogram(data = NHANES_bg, fill = "grey", alpha = .4) +
  geom_histogram(mapping = aes(fill = Gender), colour = "black") +
  facet_wrap(~ Gender, ncol = 1) +
  guides(fill = FALSE) +  # to remove the legend
  theme_bw()
```



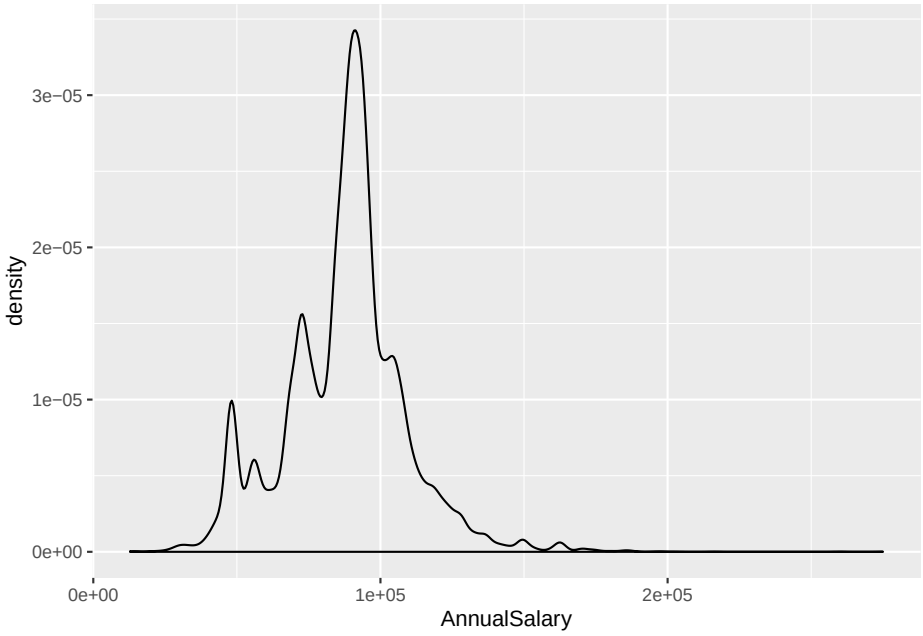
geom_boxplot: for boxplots

```
large_dept <- chi_emps[chi_emps$Department %in% c("POLICE", "FIRE", "STREETS & SAN"), ]
ggplot(data = large_dept, aes(x = Department, y = AnnualSalary)) +
  geom_boxplot()
```



geom_density: for density plots

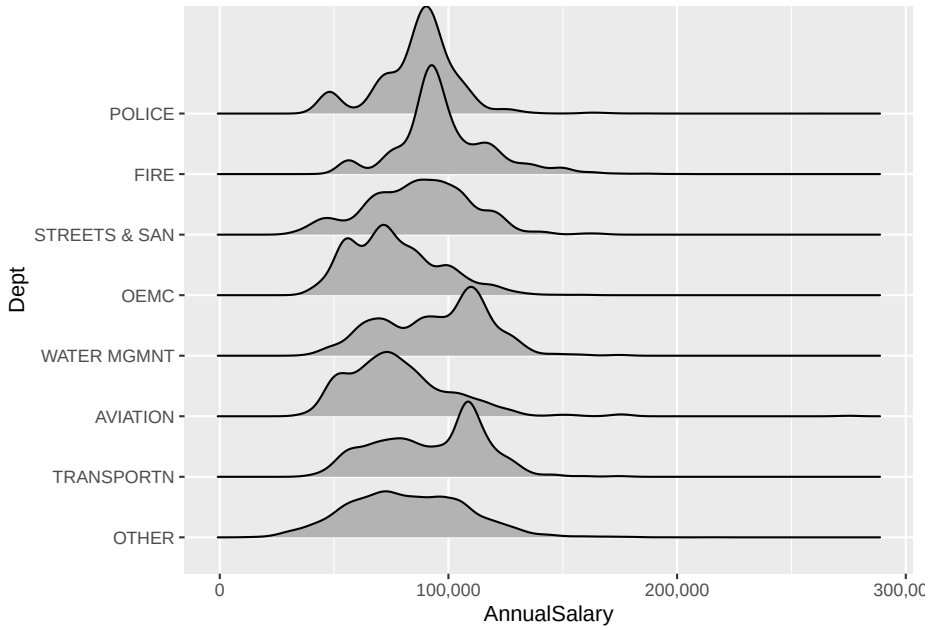
```
ggplot(data = chicago, aes(x = AnnualSalary)) + geom_density()
```



geom_density_ridges: for density plots

```
library(ggridges)

ggplot(data = chicago, aes(x = AnnualSalary, y = Dept)) +
  geom_density_ridges() +
  scale_x_continuous(label=comma)
```


POLICE
FIRE
STREETS & SAN
OEMC
WATER MGMNT
AVIATION
TRANSPORTN
OTHER
Dept
0
100,000
200,000
300,00
AnnualSalary

Chapter 11

dplyr joins

```
library(notitia)
```

```
areas
```

```
## # A tibble: 7 x 2
##   country        area
##   <chr>         <dbl>
## 1 Russia        16376
## 2 China          9388
## 3 United States  9147
## 4 Brazil         8358
## 5 India          2973
## 6 Indonesia      1811
## 7 Nigeria         910
```

```
populations
```

```
## # A tibble: 8 x 2
##   country         pop
##   <chr>         <dbl>
## 1 India          1311
## 2 United States   331
## 3 Indonesia       264
## 4 Pakistan        210
## 5 Nigeria         208
## 6 Bangladesh      161
## 7 Russia          141
## 8 Mexico          127
```

```
library(dplyr)
```

full_join

```
country_info <- full_join(areas, populations)
```

```
## Joining, by = "country"
```

```
country_info
```

```
## # A tibble: 10 x 3
##    country        area   pop
##    <chr>         <dbl> <dbl>
##  1 Russia        16376   141
##  2 China          9388    NA
##  3 United States  9147   331
##  4 Brazil         8358    NA
##  5 India          2973  1311
##  6 Indonesia      1811   264
##  7 Nigeria         910   208
##  8 Pakistan         NA   210
##  9 Bangladesh       NA   161
## 10 Mexico           NA   127
```

inner_join

```
country_info <- inner_join(areas, populations)
```

```
## Joining, by = "country"
```

```
country_info
```

```
## # A tibble: 5 x 3
##   country        area   pop
##   <chr>         <dbl> <dbl>
## 1 Russia        16376   141
## 2 United States  9147   331
## 3 India          2973  1311
## 4 Indonesia      1811   264
## 5 Nigeria         910   208
```

left_join and right_join

```
left_join(areas, populations)
```

```
## Joining, by = "country"
```

```
## # A tibble: 7 x 3
##   country        area   pop
##   <chr>         <dbl> <dbl>
## 1 Russia        16376   141
## 2 China          9388    NA
## 3 United States  9147   331
## 4 Brazil         8358    NA
## 5 India          2973  1311
## 6 Indonesia      1811   264
## 7 Nigeria         910   208
```

```
right_join(areas, populations)
```

```
## Joining, by = "country"
```

```
## # A tibble: 8 x 3
##   country        area   pop
##   <chr>         <dbl> <dbl>
## 1 India          2973  1311
## 2 United States  9147   331
## 3 Indonesia      1811   264
## 4 Pakistan         NA   210
## 5 Nigeria         910   208
## 6 Bangladesh       NA   161
## 7 Russia        16376   141
## 8 Mexico           NA   127
```

anti_join

```
anti_join(areas, populations)
```

```
## Joining, by = "country"
```

```
## # A tibble: 2 x 2
##   country  area
##   <chr>   <dbl>
## 1 China    9388
## 2 Brazil   8358
```

```
anti_join(populations, areas)
```

```
## Joining, by = "country"
```

```
## # A tibble: 3 x 2
##   country      pop
##   <chr>      <dbl>
## 1 Pakistan     210
## 2 Bangladesh   161
## 3 Mexico       127
```

Chapter 12

dplyr: Data wrangling functions

select

filter

arrange

mutate

group_by

summarise

Chapter 13

tidyr: Data wrangling functions

```
library(tidyr)
```

Splitting and combining columns

separate

```
head(lara)
```

```
## # A tibble: 6 x 8
##    Runs Inning Notout DNB   Opp      Ground  `Start Date` Match
##   <int> <fct>  <lgl>  <lgl> <chr>    <chr>   <chr>        <chr>
## 1    11 1      FALSE  FALSE Pakistan Karachi 9-Nov-90     ODI # 639
## 2    44 1      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 3     5 2      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 4    23 1      FALSE  FALSE England  Lord's  27-May-91    ODI # 678
## 5     5 1      FALSE  FALSE Pakistan Sharjah 17-Oct-91    ODI # 679
## 6    45 1      FALSE  FALSE India    Sharjah 19-Oct-91    ODI # 681
```

```
lara2 <- separate(lara, Match, into = c("Format", "MatchNum"), sep = " # " )
head(lara2)
```

```
## # A tibble: 6 x 9
##    Runs Inning Notout DNB   Opp      Ground  `Start Date` Format MatchNum
##   <int> <fct>  <lgl>  <lgl> <chr>    <chr>   <chr>        <chr>  <chr>
## 1    11 1      FALSE  FALSE Pakistan Karachi 9-Nov-90     ODI    639
## 2    44 1      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test   1158
## 3     5 2      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test   1158
## 4    23 1      FALSE  FALSE England  Lord's  27-May-91    ODI    678
## 5     5 1      FALSE  FALSE Pakistan Sharjah 17-Oct-91    ODI    679
## 6    45 1      FALSE  FALSE India    Sharjah 19-Oct-91    ODI    681
```

unite

```
lara3 <- unite(lara2, col = Match, Format, MatchNum, sep = " # " )
head(lara3)
```

```
## # A tibble: 6 x 8
##    Runs Inning Notout DNB   Opp      Ground  `Start Date` Match
##   <int> <fct>  <lgl>  <lgl> <chr>    <chr>   <chr>        <chr>
## 1    11 1      FALSE  FALSE Pakistan Karachi 9-Nov-90     ODI # 639
## 2    44 1      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 3     5 2      FALSE  FALSE Pakistan Lahore  6-Dec-90     Test # 1158
## 4    23 1      FALSE  FALSE England  Lord's  27-May-91    ODI # 678
## 5     5 1      FALSE  FALSE Pakistan Sharjah 17-Oct-91    ODI # 679
## 6    45 1      FALSE  FALSE India    Sharjah 19-Oct-91    ODI # 681
```

```
lara4 <- unite(lara2, col = Match, Format, MatchNum, sep = " # ", remove = FALSE )
head(lara4)
```

```
## # A tibble: 6 x 10
##    Runs Inning Notout DNB   Opp   Ground `Start Date` Match Format MatchNum
##   <int> <fct>  <lgl>  <lgl> <chr> <chr>  <chr>        <chr> <chr>  <chr>
## 1    11 1      FALSE  FALSE Paki~ Karac~ 9-Nov-90     ODI ~ ODI    639
## 2    44 1      FALSE  FALSE Paki~ Lahore 6-Dec-90     Test~ Test   1158
## 3     5 2      FALSE  FALSE Paki~ Lahore 6-Dec-90     Test~ Test   1158
## 4    23 1      FALSE  FALSE Engl~ Lord's 27-May-91    ODI ~ ODI    678
## 5     5 1      FALSE  FALSE Paki~ Sharj~ 17-Oct-91    ODI ~ ODI    679
## 6    45 1      FALSE  FALSE India Sharj~ 19-Oct-91    ODI ~ ODI    681
```

Reshaping data

```
unemp
```

```
## # A tibble: 72 x 13
##     Year   Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  1948   3.4   3.8   4     3.9   3.5   3.6   3.6   3.9   3.8   3.7   3.8
##  2  1949   4.3   4.7   5     5.3   6.1   6.2   6.7   6.8   6.6   7.9   6.4
##  3  1950   6.5   6.4   6.3   5.8   5.5   5.4   5     4.5   4.4   4.2   4.2
##  4  1951   3.7   3.4   3.4   3.1   3     3.2   3.1   3.1   3.3   3.5   3.5
##  5  1952   3.2   3.1   2.9   2.9   3     3     3.2   3.4   3.1   3     2.8
##  6  1953   2.9   2.6   2.6   2.7   2.5   2.5   2.6   2.7   2.9   3.1   3.5
##  7  1954   4.9   5.2   5.7   5.9   5.9   5.6   5.8   6     6.1   5.7   5.3
##  8  1955   4.9   4.7   4.6   4.7   4.3   4.2   4     4.2   4.1   4.3   4.2
##  9  1956   4     3.9   4.2   4     4.3   4.3   4.4   4.1   3.9   3.9   4.3
## 10  1957   4.2   3.9   3.7   3.9   4.1   4.3   4.2   4.1   4.4   4.5   5.1
## # ... with 62 more rows, and 1 more variable: Dec <dbl>
```

gather

```
unemp2 <- gather(unemp, key = Month, value = Rate, -Year)
unemp2
```

```
## # A tibble: 864 x 3
##     Year Month  Rate
##    <dbl> <chr> <dbl>
##  1  1948 Jan     3.4
##  2  1949 Jan     4.3
##  3  1950 Jan     6.5
##  4  1951 Jan     3.7
##  5  1952 Jan     3.2
##  6  1953 Jan     2.9
##  7  1954 Jan     4.9
##  8  1955 Jan     4.9
##  9  1956 Jan     4
## 10  1957 Jan     4.2
## # ... with 854 more rows
```

```
unemp3 <- gather(unemp, key = Month, value = Rate, `Jan`:`Dec`)
unemp3
```

```
## # A tibble: 864 x 3
##     Year Month  Rate
##    <dbl> <chr> <dbl>
##  1  1948 Jan     3.4
##  2  1949 Jan     4.3
##  3  1950 Jan     6.5
##  4  1951 Jan     3.7
##  5  1952 Jan     3.2
##  6  1953 Jan     2.9
##  7  1954 Jan     4.9
##  8  1955 Jan     4.9
##  9  1956 Jan     4
## 10  1957 Jan     4.2
## # ... with 854 more rows
```

```
unemp4 <- gather(unemp, key = Month, value = Rate, 2:12)
unemp4
```

```
## # A tibble: 792 x 4
##     Year   Dec Month  Rate
##    <dbl> <dbl> <chr> <dbl>
##  1  1948   4   Jan     3.4
##  2  1949   6.6 Jan     4.3
##  3  1950   4.3 Jan     6.5
##  4  1951   3.1 Jan     3.7
##  5  1952   2.7 Jan     3.2
##  6  1953   4.5 Jan     2.9
##  7  1954   5   Jan     4.9
##  8  1955   4.2 Jan     4.9
##  9  1956   4.2 Jan     4
## 10  1957   5.2 Jan     4.2
## # ... with 782 more rows
```

spread

```
spread(unemp2, key = Month, value = Rate)
```

```
## # A tibble: 72 x 13
##     Year   Apr   Aug   Dec   Feb   Jan   Jul   Jun   Mar   May   Nov   Oct
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1  1948   3.9   3.9   4     3.8   3.4   3.6   3.6   4     3.5   3.8   3.7
##  2  1949   5.3   6.8   6.6   4.7   4.3   6.7   6.2   5     6.1   6.4   7.9
```

```
##  3  1950   5.8   4.5   4.3   6.4   6.5   5     5.4   6.3   5.5   4.2   4.2
##  4  1951   3.1   3.1   3.1   3.4   3.7   3.1   3.2   3.4   3     3.5   3.5
##  5  1952   2.9   3.4   2.7   3.1   3.2   3.2   3     2.9   3     2.8   3
##  6  1953   2.7   2.7   4.5   2.6   2.9   2.6   2.5   2.6   2.5   3.5   3.1
##  7  1954   5.9   6     5     5.2   4.9   5.8   5.6   5.7   5.9   5.3   5.7
##  8  1955   4.7   4.2   4.2   4.7   4.9   4     4.2   4.6   4.3   4.2   4.3
##  9  1956   4     4.1   4.2   3.9   4     4.4   4.3   4.2   4.3   4.3   3.9
## 10  1957   3.9   4.1   5.2   3.9   4.2   4.2   4.3   3.7   4.1   5.1   4.5
## # ... with 62 more rows, and 1 more variable: Sep <dbl>
```

Chapter 14

Intro Statistical functions

sample

set.seed

Chapter 15

Data sets in the notitia package

unemp

Historical unemployment rates in the United States ## chi_emps{-#chi} Human Resources data for all employees of the city of Chicago, Illinois (USA) as of April 2019. ## populations {-#populations} Population data (in millions) for some of the world's largest countries. ## areas {-#areas} Areas in square for ## complete_populations {-#comppops}

Population data (in millions) for some of the world's largest countries. This data is similar to that in **populations** but contains some additional entries.

complete_areas

This data is similar to that in **areas** but contains some additional entries.

capitals

Table containing the capitals of 10 countries.

lebron

Career regular-season statistics of NBA player, LeBron James.

jordan

Career regular-season statistics of NBA player, Michael Jordan.

nyc_sat10

Performance of NYC public schools on the SAT exam in 2010.

nyc_sat12

Performance of NYC public schools on the SAT exam in 2012.

apple_prod

Quarterly sales data published by Apple Inc for various product lines.

flight_data

Flight information for Delta Airlines flights in 2016

rafa_novak

Tennis matches played between Rafael Nadal and Novak Djokovic.

lara

Career batting statistics of Brian Lara, West Indian cricketer.

electricity

Electricity consumption by country for the period 2008 to 2018.