PS 312: Programming with R
Course Notes

Abhijit Dasgupta, PhD

Last updated: March 27, 2019

Welcome

This course is an introduction to the statistical programming language R and various applications. We will cover the entire data analytics pipeline from data ingestion to data wrangling, summarizing, modeling, visualizing and reporting, all using tools found within the R ecosystem.

The version of these notes you are reading now was built on 2019-03-27.

Reproducibility

These notes are written with `bookdown`, a R package for writing books using `rmarkdown`. All code in these notes were developed on R version 3.5.0 (2018-04-23), using the same packages pre-installed in your virtual machines. When you're on your own, you will need to install a recent version of R, and also install the corresponding packages, on your computer, for all the code to work. A listing of all the packages used in this course will be available as an appendix.

To build these notes locally, clone or download the Github repo hosting these notes, unzip it if necessary, and double-click on `FSI_Book.Rproj`. Assuming you have RStudio installed, this will open this project (more on *RStudio Projects* later). You can then go to the console and enter the following code:

```r
bookdown::render_book("index.Rmd") # to build these notes
browseURL("_book/index.html") # to view it
```

Chapter 1

Data visualization

1.1 ggplot2

We'll be primarily using ggplot2 in this workshop.

- Makes pretty good formatting choices out of the box
- Works like pipes!!
- Is declarative (tell it what you want) without getting caught up in minutae
- Strongly leverages data frames (good practice)
- Fast enough
- There are good templates if you want to change the look

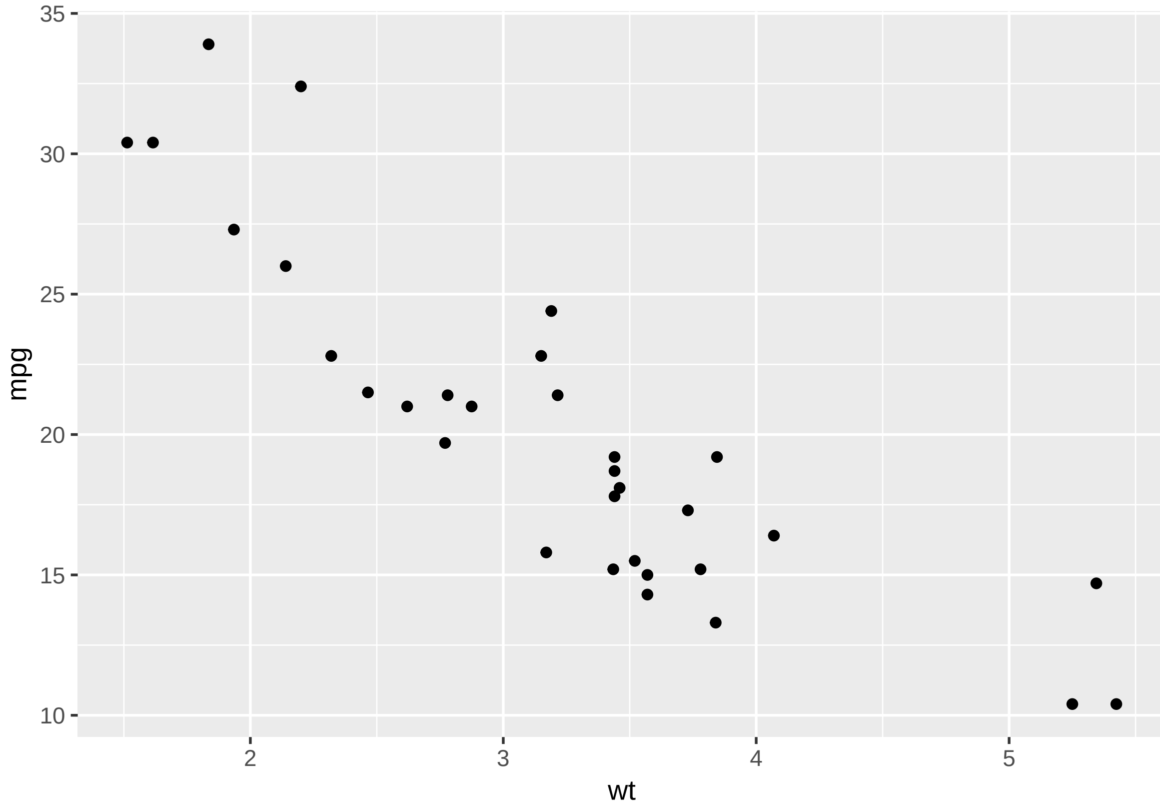
The `ggplot2` package is a very flexible and (to me) intuitive way of visualizing data. It is based on the concept of layering elements on a canvas.

> This idea of layering graphics on a canvas is, to me, a nice way of building graphs

- A `data.frame` object
- *Aesthetic mappings* (aes) to say what data is used for what purpose in the viz
  - x- and y-direction
  - shapes, colors, lines
- A *geometry object* (geom) to say what to draw
  - You can "layer" geoms on each other to build plots

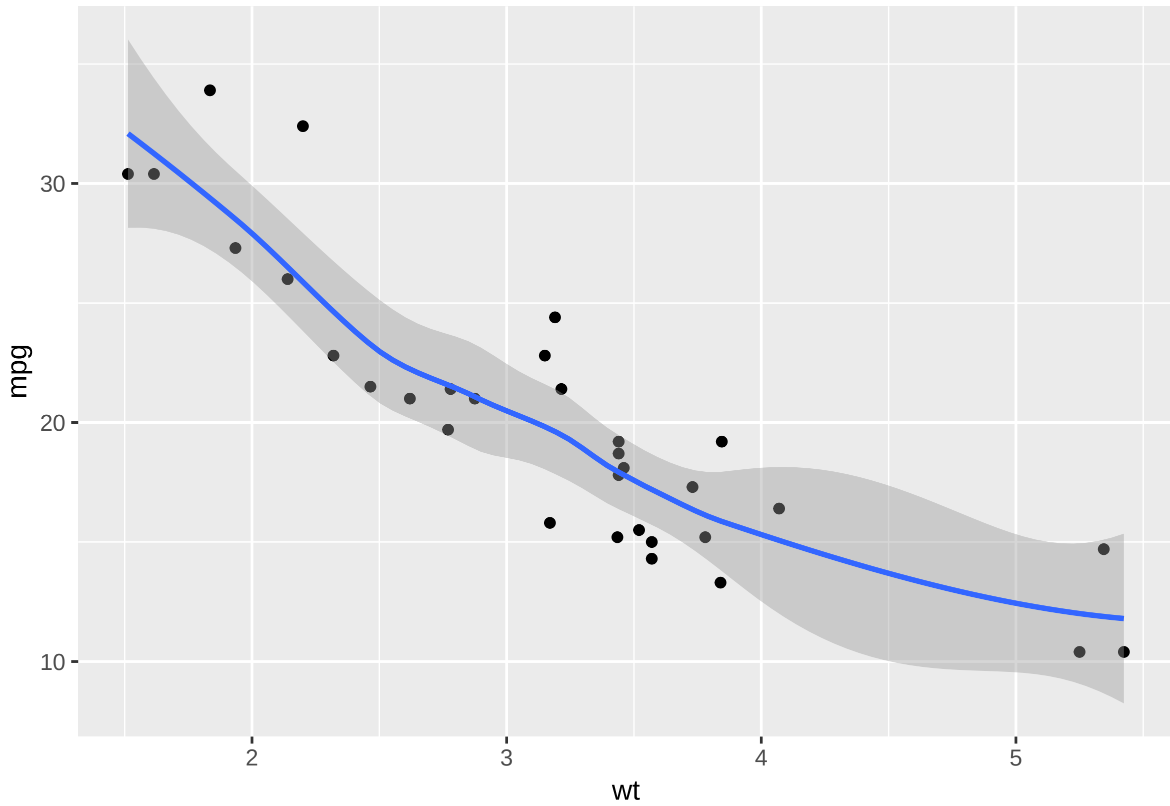
> `ggplot` used pipes before pipes were a thing.
>
> However, it uses the `+` symbol for piping rather than the `%>%` operator, since it pre-dates the `tidyverse`

```
library(ggplot2)
ggplot(mtcars, aes(x = wt, y = mpg)) + geom_point()
```



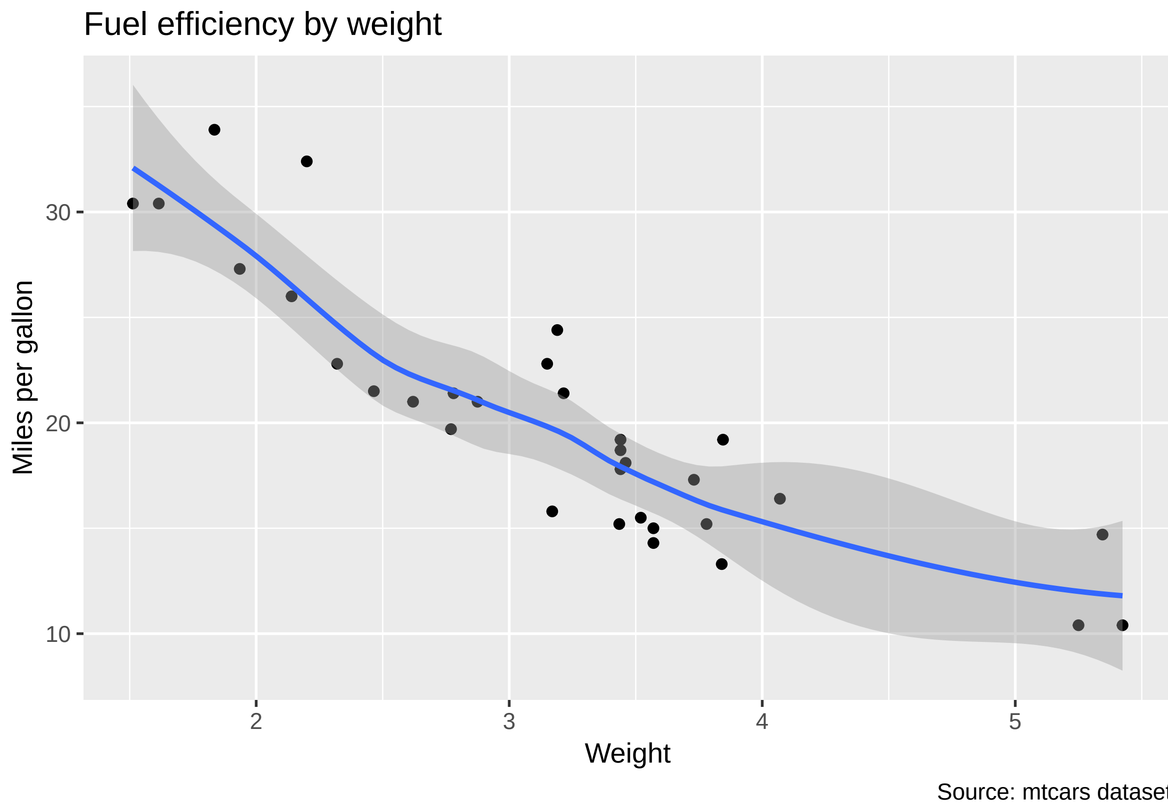
- A `data.frame` object: mtcars
- Aesthetic mapping:
  - x-axis: wt
  - y-axis: mpg
- Geometry:
  - geom_point: draw points

```
ggplot(mtcars, aes(x = wt, y = mpg)) +
  geom_point()+
  geom_smooth()
```



- A `data.frame` object: mtcars
- Aesthetic mapping:
  - x-axis: wt
  - y-axis: mpg
- Geometry:
  - geom_point: draw points
  - geom_smooth: Add a layer which draws a best-fitting line

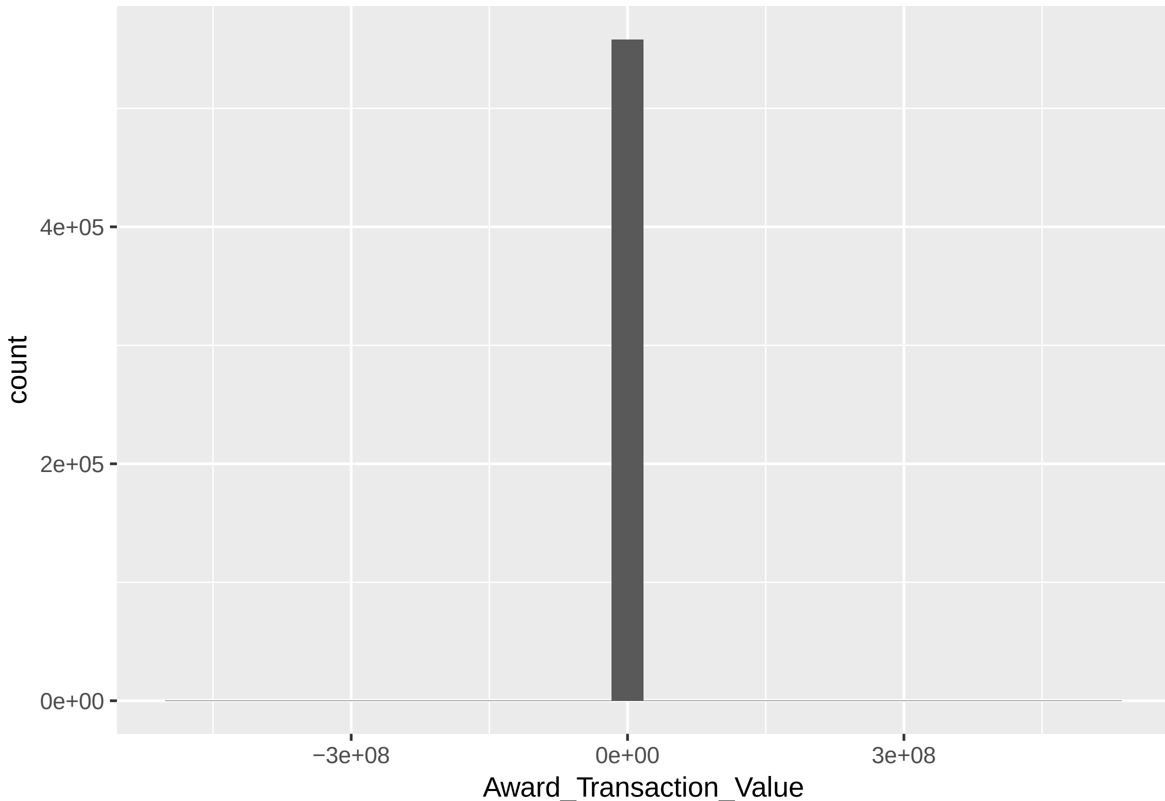
Now we clean up the plot to make it presentable.



1.2 Single continuous variable

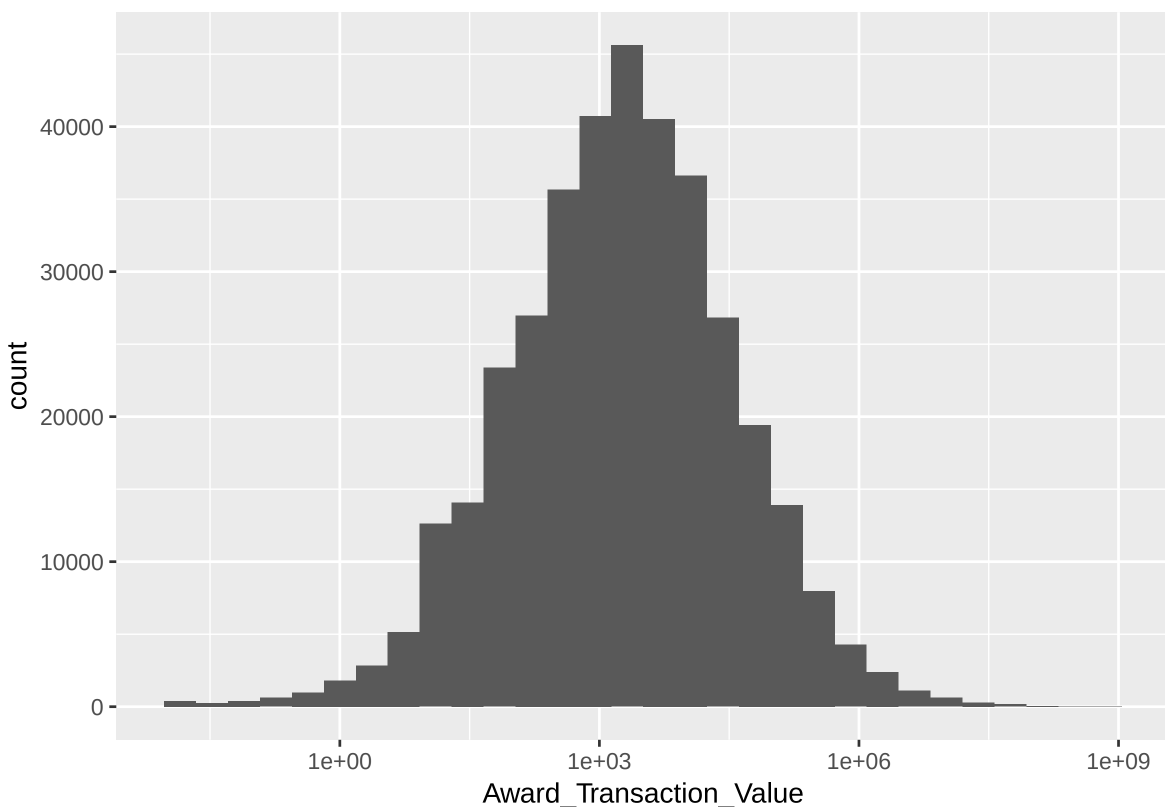
1.2.1 Histogram

```
dos <- import('data/Department of State.csv')
dos %>%
  ggplot(aes(x = Award_Transaction_Value)) + geom_histogram()
```



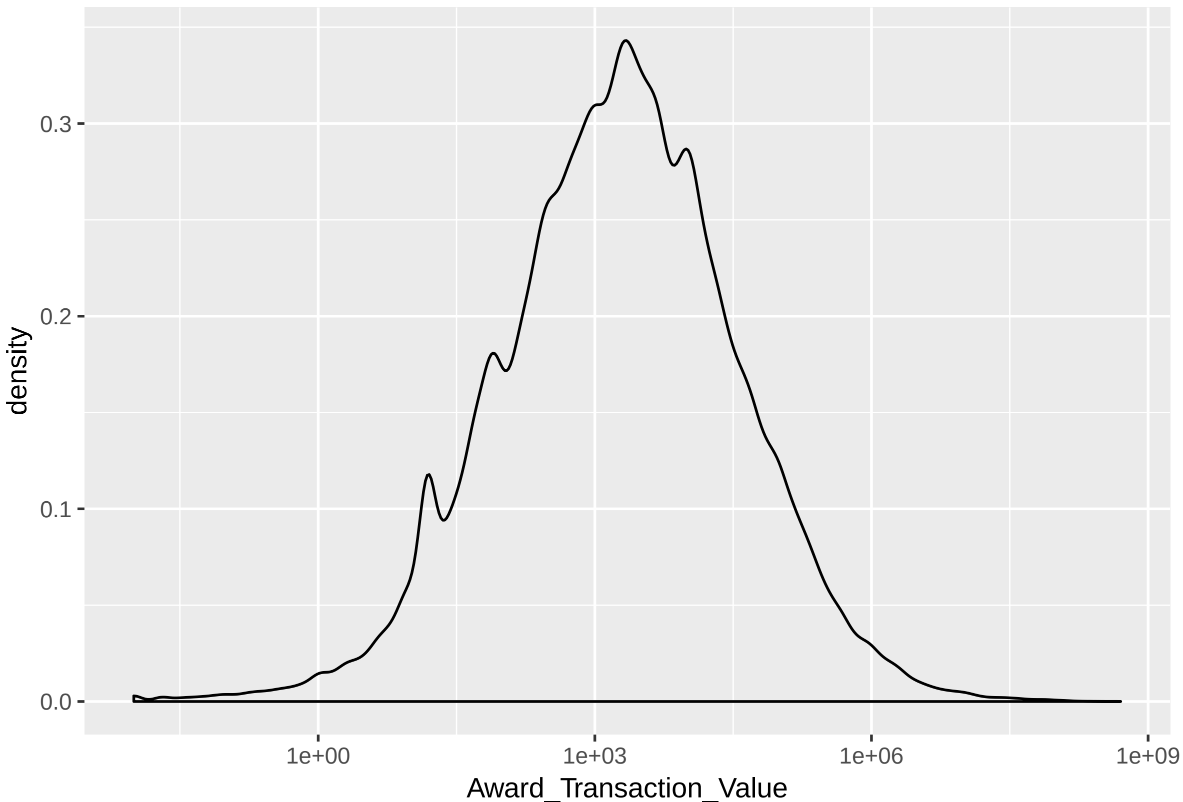
Change the axis to the log scale for better visual

```
dos %>%
  ggplot(aes(x = Award_Transaction_Value)) + geom_histogram()+
  scale_x_log10() # x-axis on log scale
```



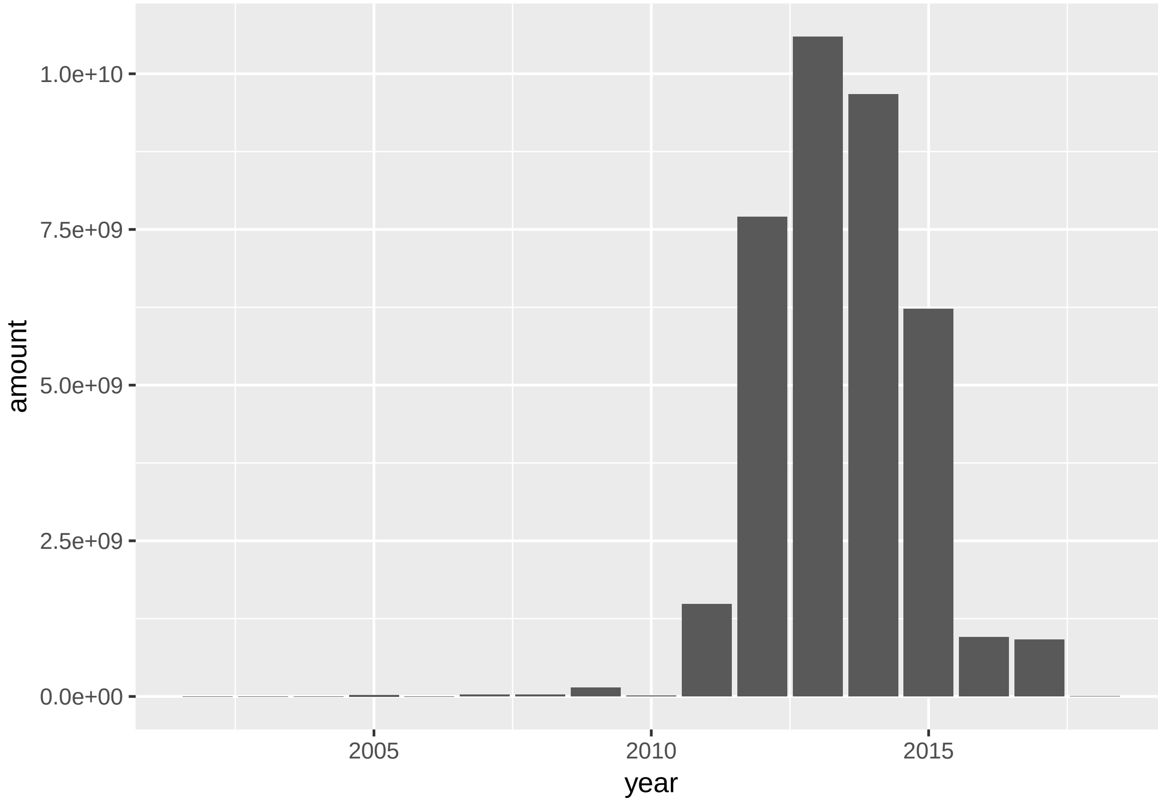
1.2.2 Density plot

```
dos %>%
  ggplot(aes(x = Award_Transaction_Value)) + geom_density()+
  scale_x_log10()
```



1.3 Bar plots

```
library(lubridate)
dos %>%
  group_by(year = year(as_date(Award_Start_Date))) %>%
  summarize(amount = sum(Award_Transaction_Value)) %>%
  ggplot(aes(x = year, y = amount)) + # Note change in pipe operator
    geom_bar(stat='identity')
```

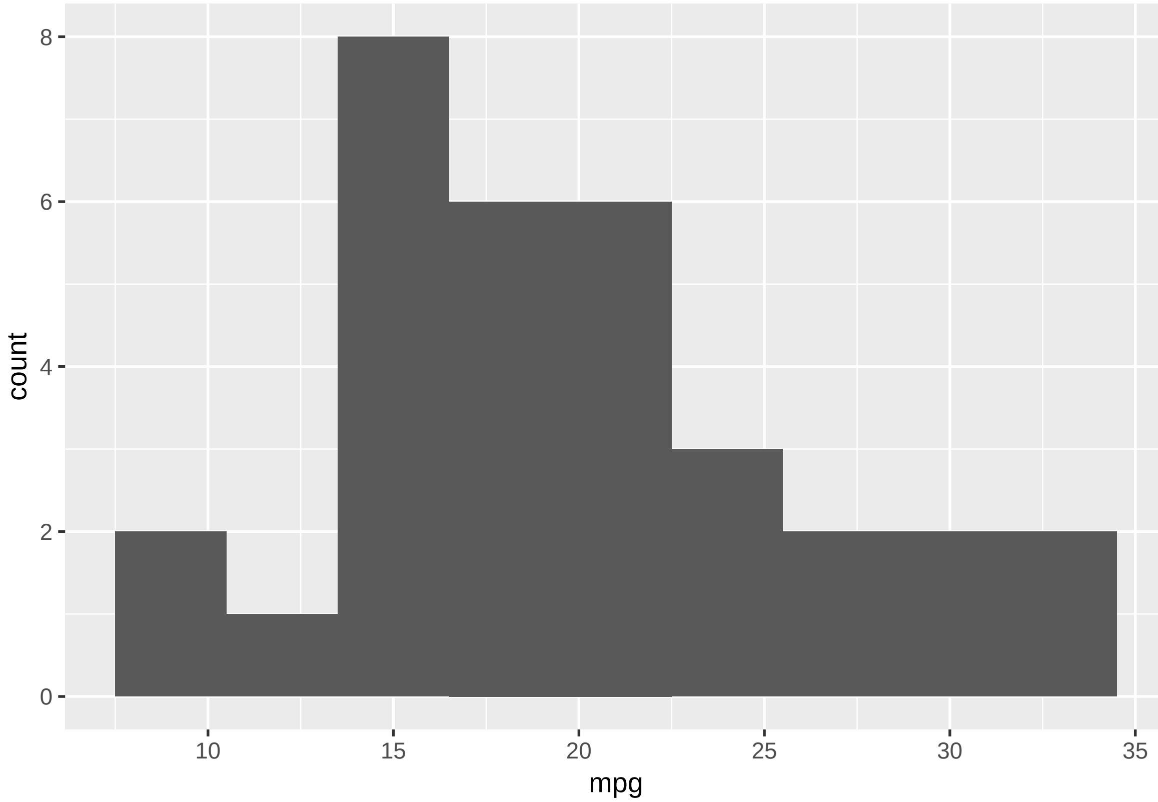


1.3.1 Exercise

Using the `mtcars` dataset in R, create:

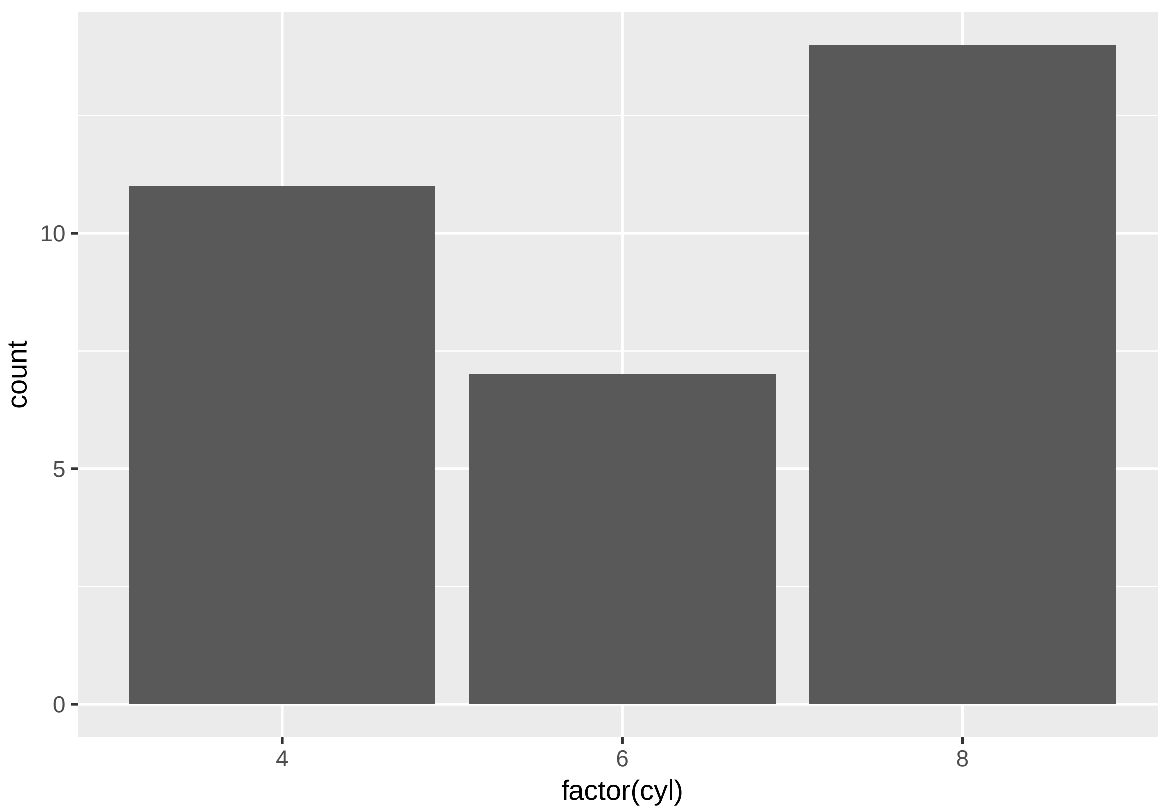
1. A histogram of the fuel efficiences (`mpg`) in the data set
2. A bar plot of frequencies of number of cylinders (`cyl`) in the car

```
ggplot(mtcars, aes(x = mpg)) + geom_histogram(binwidth=3)
```




```r
# ggplot(mtcars) + geom_histogram(aes(x = mpg), binwidth = 3)
```

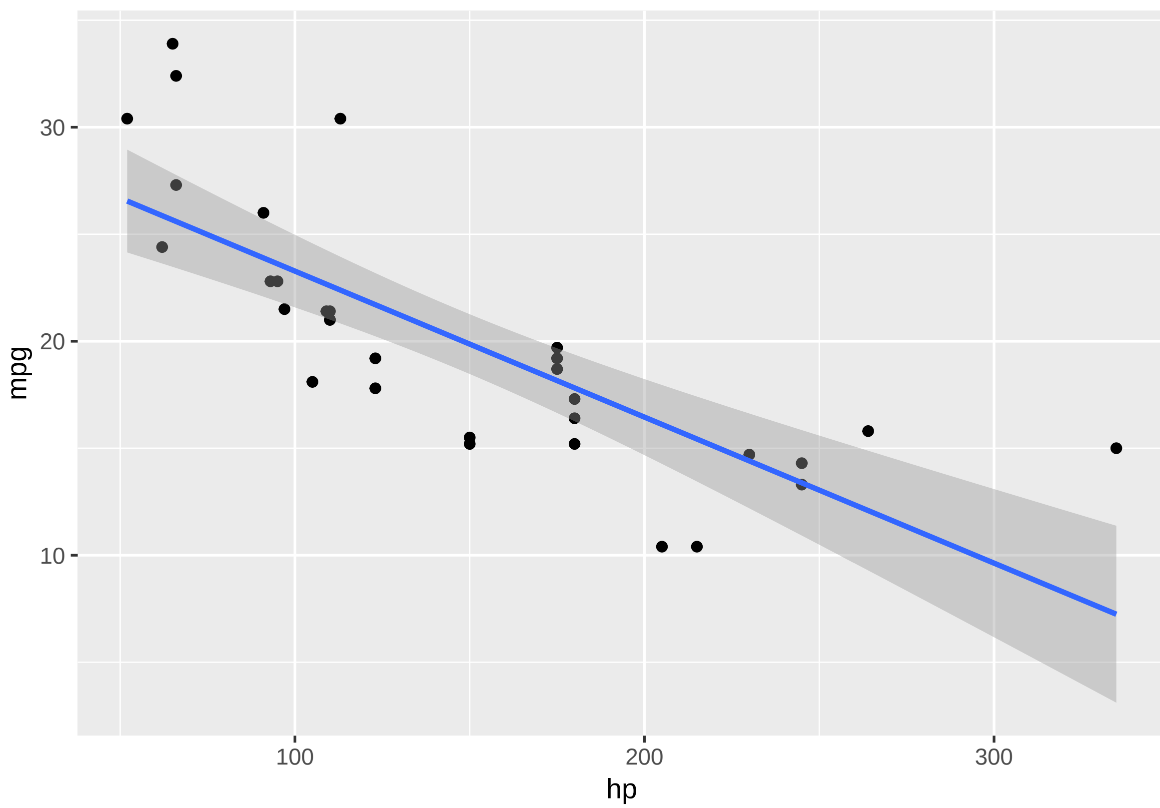
```r
ggplot(mtcars, aes(x = factor(cyl))) + geom_bar()
```



1.4 Two continuous variables

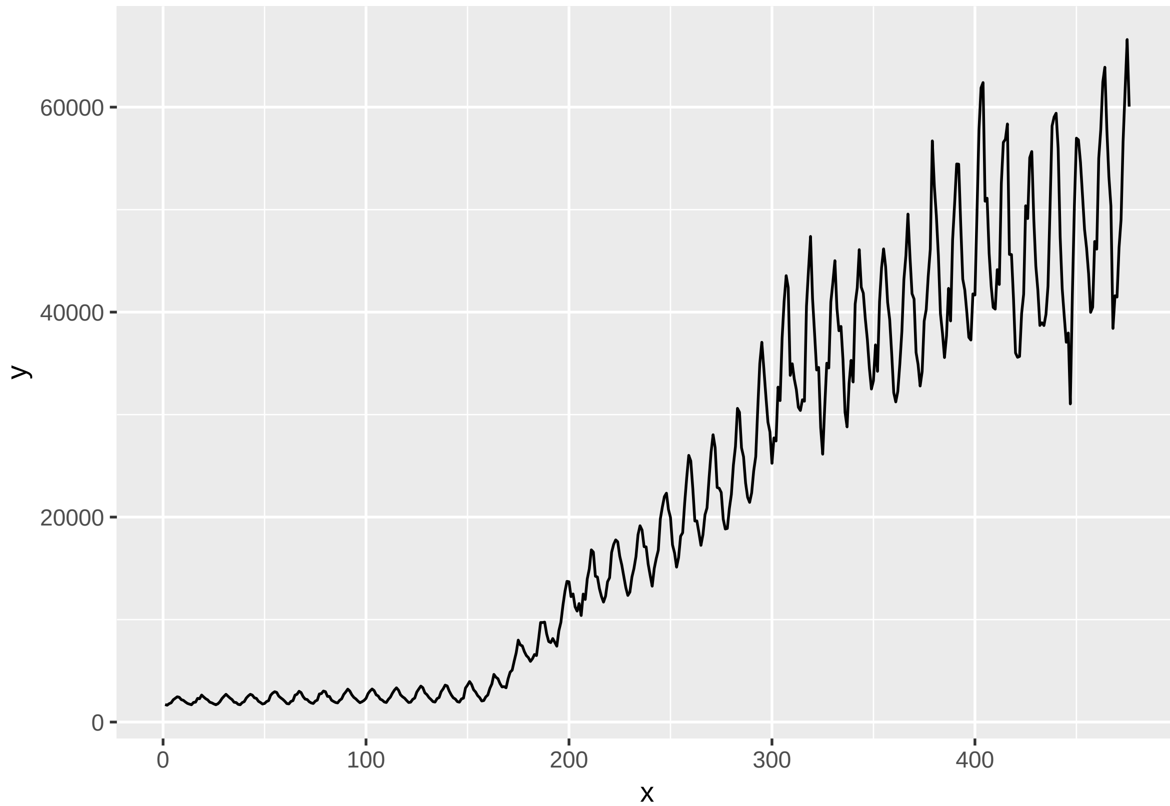
1.4.1 Adding a best fitting straight line

```
ggplot(mtcars, aes(x = hp, y = mpg))+
  geom_point()+
  geom_smooth(method = 'lm')
```



1.5 Time series

```
library(forecast)
d <- data.frame(x = 1:length(gas), y = gas) # Australian monthly gas production
ggplot(d, aes(x, y)) + geom_line()
```



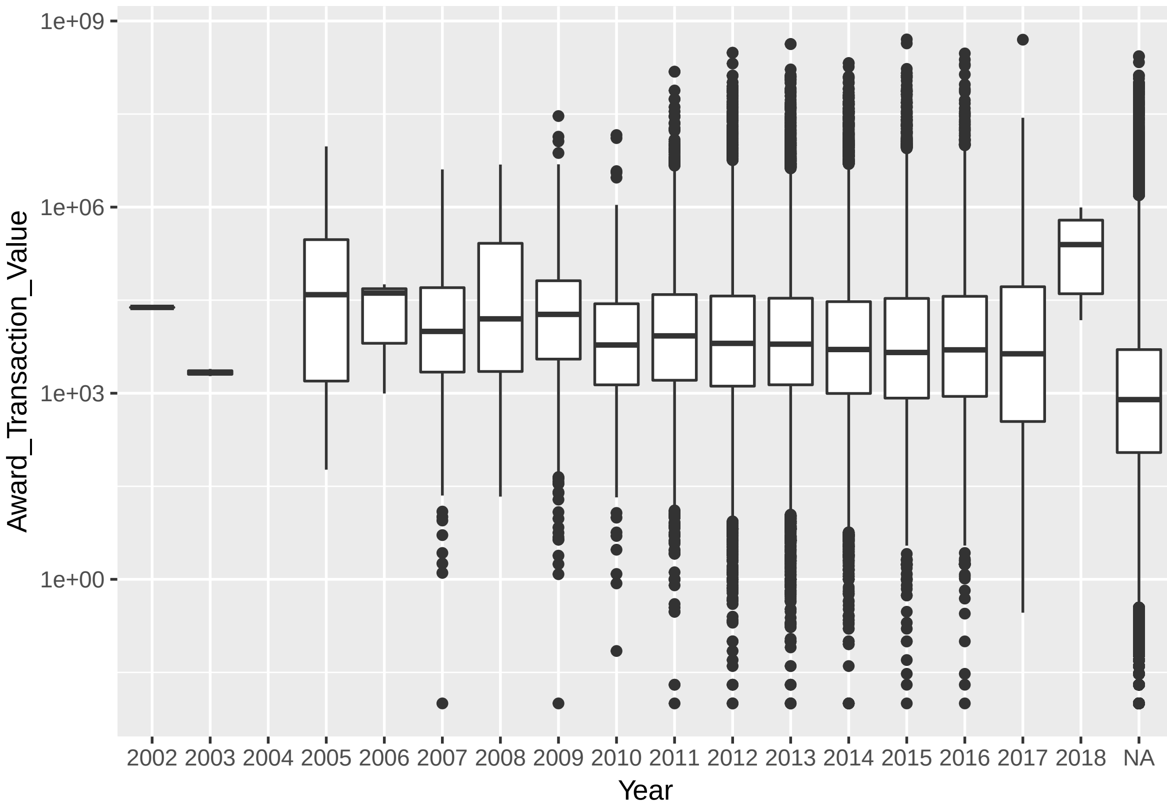
1.5.1 Exercise

1. Create a scatter plot of sepal length and sepal width from the `iris` dataset, and add a smooth line through it

1.6 Continuous variable with discrete variable

1.6.1 Boxplot

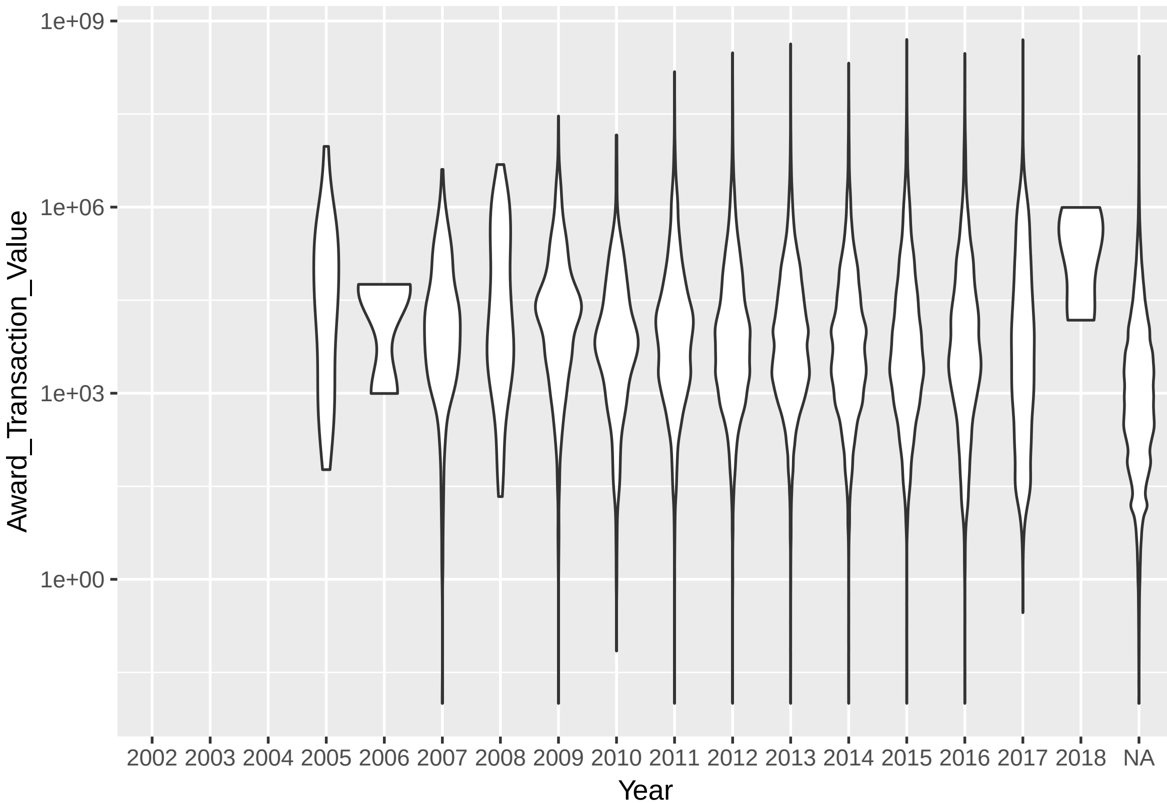
```
dos %>%
ggplot(aes(x = factor(year(as_date(Award_Start_Date))),
                y = Award_Transaction_Value))+
  geom_boxplot() +
  scale_y_log10()+
  labs(x = 'Year')
```



1.6.2 Violin plot

This is essetially a reflected density plot and gives a better sense of the data distribution

```
dos %>%
ggplot(aes(x = factor(year(as_date(Award_Start_Date))),
                y = Award_Transaction_Value))+
  geom_violin() +
  scale_y_log10()+
  labs(x = 'Year')
```



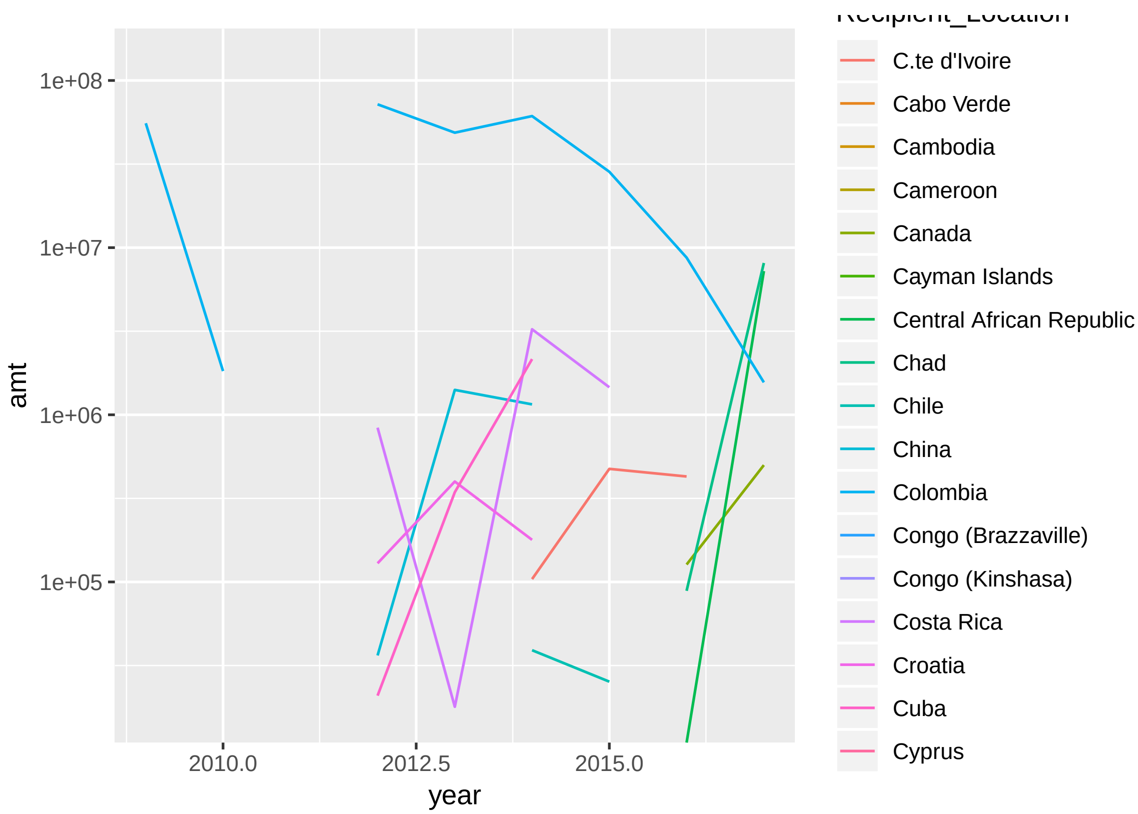
1.6.3 Exercise

1. Plot a boxplot of petal length by species using the `iris` dataset

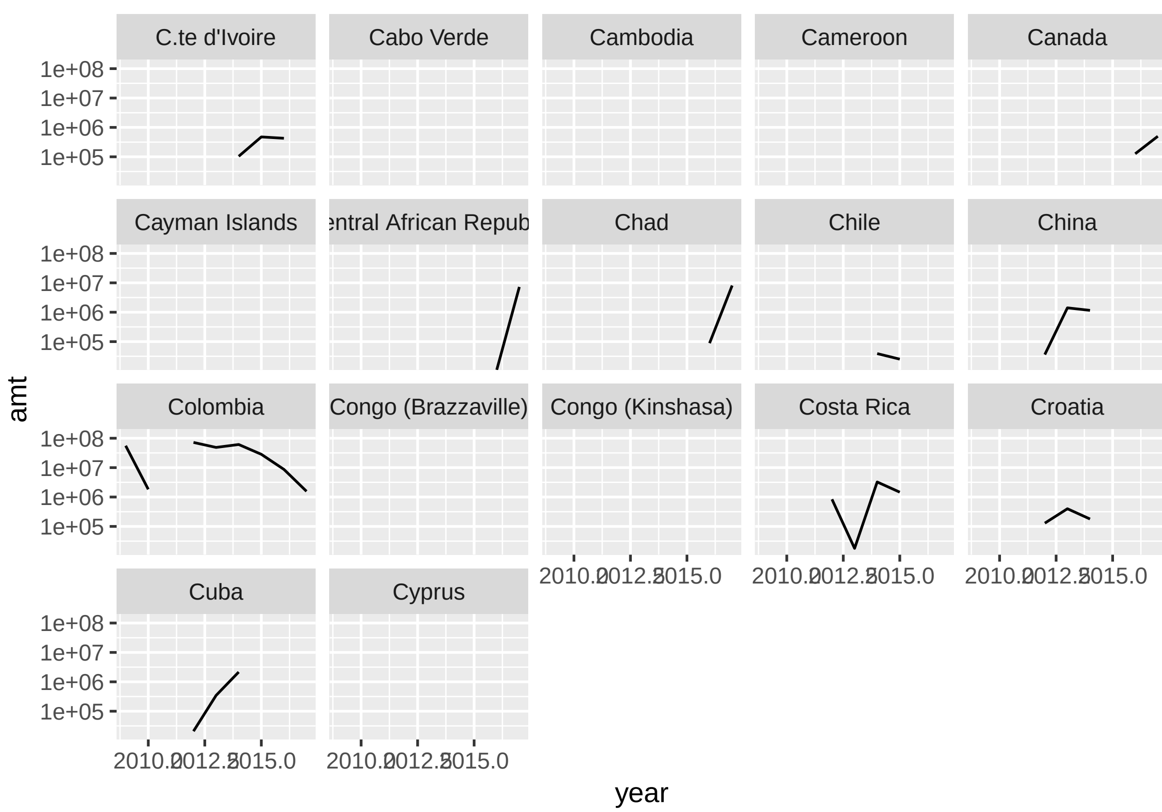
1.7 Grouped visualizations

We're going to plot the change in aid provided to each country over time. To do this we need summaries by time and location

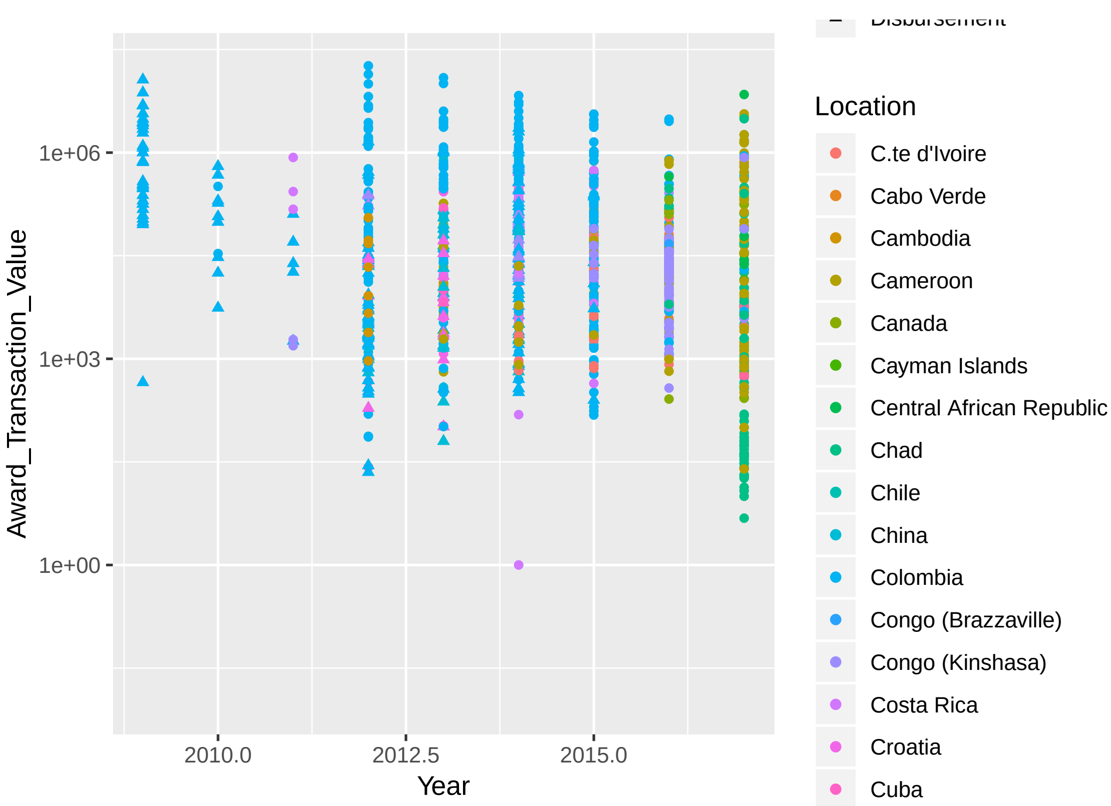
```
grp_data <- dos %>%
  group_by(Recipient_Location, year = year(as_date(Award_Start_Date))) %>%
  summarize(amt = sum(Award_Transaction_Value)) %>%
  filter(str_detect(Recipient_Location, '^C'))
ggplot(grp_data, aes(x = year, y = amt, color=Recipient_Location))+
  geom_line()+
  scale_y_log10()
```



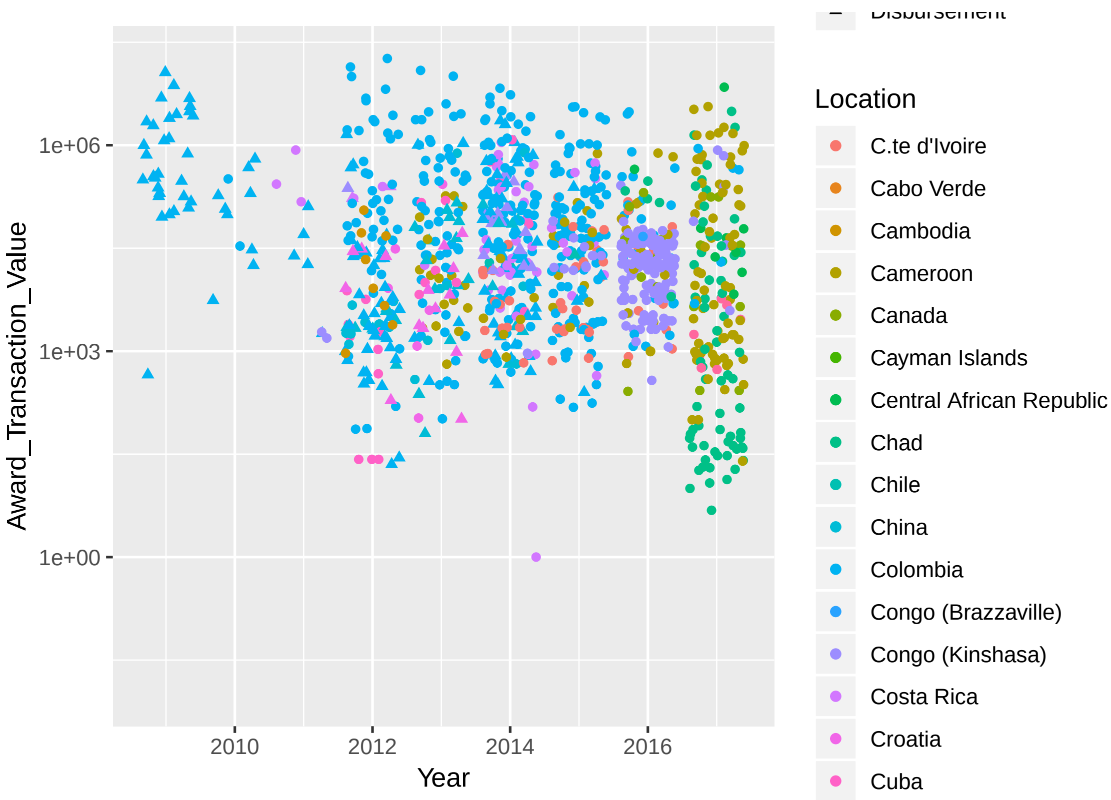

```r
ggplot(grp_data, aes(x = year,  y = amt))+
  geom_line()+
  scale_y_log10()+
  facet_wrap(~Recipient_Location)
```



```
## dos %>% filter(str_detect(Recipient_Location, '^C')) %>%
## ggplot(aes(x = year(as_date(Award_Start_Date)),
##             y = Award_Transaction_Value,
##             color = Recipient_Location,
##             shape = Award_Transaction_Type))+
##   geom_point()+
##   labs(x = 'Year', color='Location')+
##   scale_y_log10()
##
```

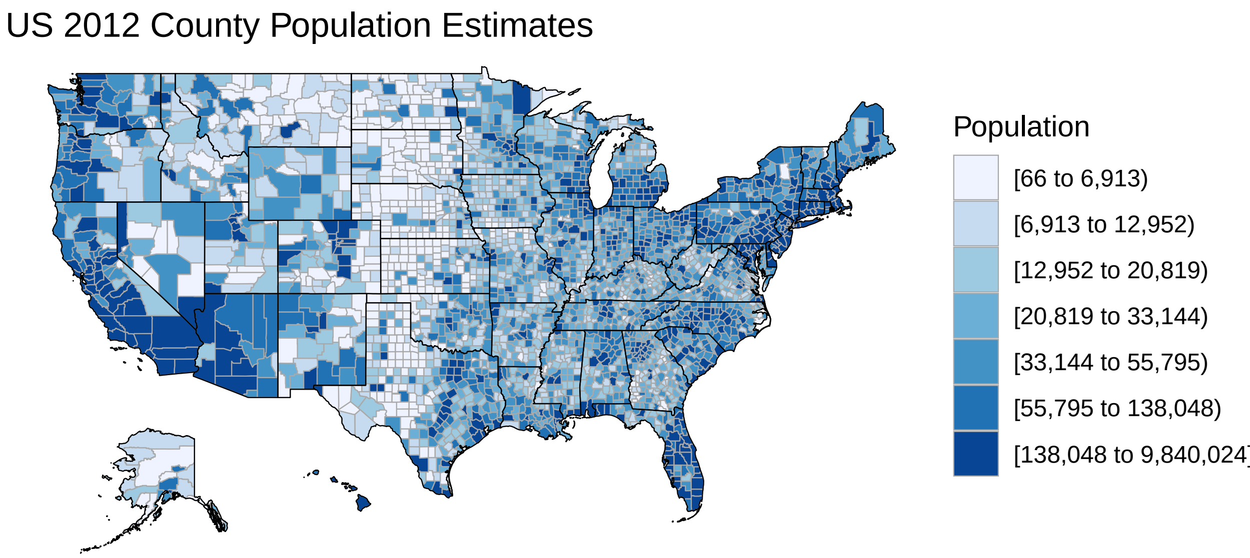



```
## dos %>% filter(str_detect(Recipient_Location, '^C')) %>%
## ggplot(aes(x = year(as_date(Award_Start_Date)),
##             y = Award_Transaction_Value,
##             color = Recipient_Location,
##             shape = Award_Transaction_Type))+
##   geom_jitter()+
##   labs(x = 'Year', color='Location')+
##   scale_y_log10()
##
```




```
schools <- rio::import('data/schools.rds')
schools %>% filter(tophead=='Elementary schools',
                   head2=="Average hours in school day") %>%
  filter(!is.na(State), State != 'United States') %>%
  ggplot(aes(x = State, y = stats, ymin = stats - 2*se,
             ymax = stats + 2*se)) +
  geom_pointrange()+
  labs(y = 'Avg hours in school day')+
  theme_bw()+
  theme(axis.text.x = element_text(angle=45, hjust = 1))
```

1.8 Maps



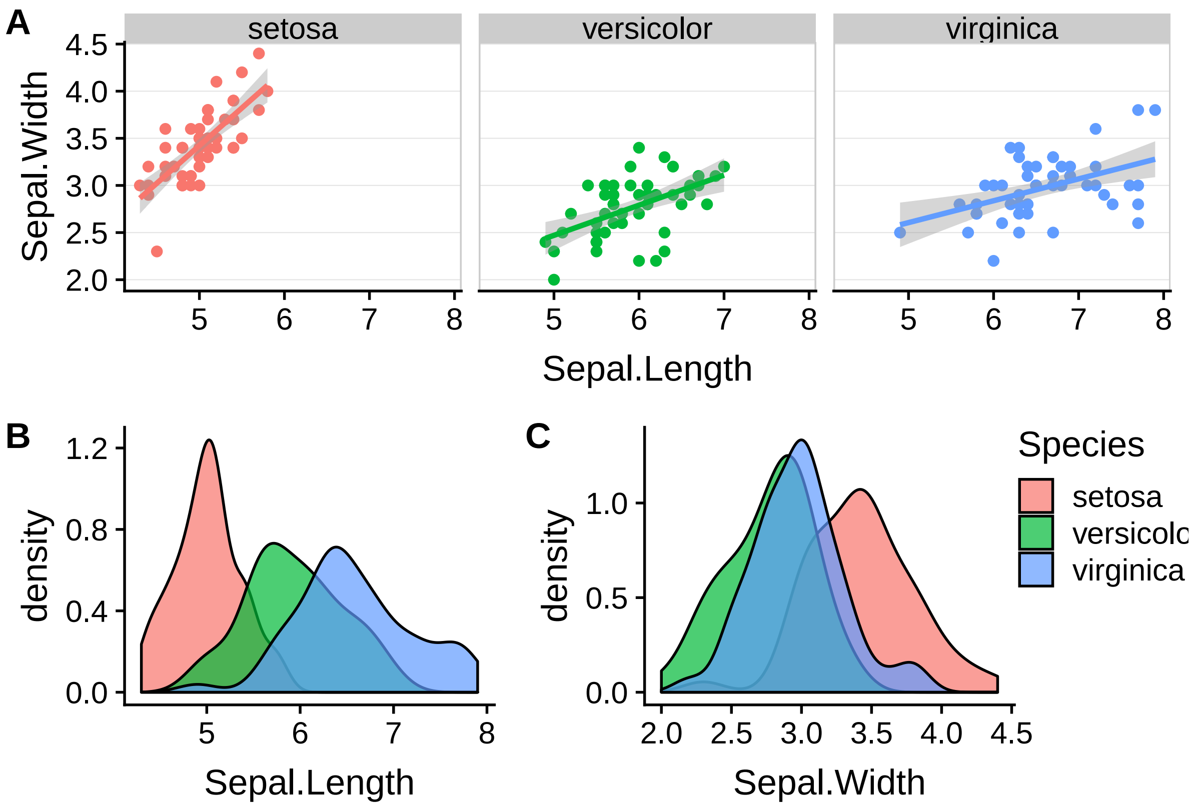
We can also ingest SHP files to draw maps. We don't show the final version since it took too long to render.

```
library(sf)
hrr_info <- st_read('~/Downloads/hrr_bdry-1/HRR_Bdry.SHP')
head(hrr_info)
ggplot(hrr_info)+geom_sf()
ggsave('map.png')
```

1.9 Stitching graphs together.

```
# install.packages('cowplot')
library(cowplot)
p1 <- ggplot(iris, aes(Sepal.Length, Sepal.Width, color = Species)) +
 geom_point() + facet_grid(. ~ Species) + stat_smooth(method = "lm") +
 background_grid(major = 'y', minor = "none") +
 panel_border() + theme(legend.position = "none")

# plot B
p2 <- ggplot(iris, aes(Sepal.Length, fill = Species)) +
 geom_density(alpha = .7) + theme(legend.justification = "top")
p2a <- p2 + theme(legend.position = "none")

# plot C
p3 <- ggplot(iris, aes(Sepal.Width, fill = Species)) +
 geom_density(alpha = .7) + theme(legend.position = "none")

# legend
legend <- get_legend(p2)
```

```r
# align all plots vertically
plots <- align_plots(p1, p2a, p3, align = 'v', axis = 'l')

# put together bottom row and then everything
bottom_row <- plot_grid(plots[[2]], plots[[3]], legend, labels = c("B", "C"), rel_width
plot_grid(plots[[1]], bottom_row, labels = c("A"), ncol = 1)
```




```r
## library(ggplot2)
## library(plotly)
## p=ggplot(iris, aes(x=Sepal.Length,
##                    y=Sepal.Width,
##                    color=Species,
##                    shape=Species)) +
##     geom_point(size=6, alpha=0.6)
## mytext=paste("Sepal Length = ", iris$Sepal.Length,
##              "\n" , "Sepal Width = ", iris$Sepal.Width,
##              "\n", "Row Number: ",rownames(iris),  sep="")
## pp=plotly::plotly_build(p)
## style( pp, text=mytext,
##        hoverinfo = "text",
##        traces = c(1, 2, 3) )
```

1.10 Interactive graphics

We won't put these in the notes, since they don't work well in printed form

Chapter 2

Functions

```r
myDumbFunction <- function() 42
myDumbFunction()
```

```
[1] 42
```

```r
doubleIt <- function(x) {
  myResult <- x * 2
  myResult # or, explicitly, return(myResult)
}
doubleIt(5)
```

```
[1] 10
```

```r
exists("myResult")
```

```
[1] FALSE
```

```r
myResult <- 1000
doubleItOutput <- doubleIt(2)
myResult
```

```
[1] 1000
```

```r
my_sum <- function(x){
  s <- sum(x)
  n <- length(x)
  result <- s / n
  return(result)
}
```

```r
my_sum(1:10)
```

```
[1] 5.5
```

```r
answer <- my_sum(1:10)
answer
```

```
[1] 5.5
```

```r
my_sum <- function(x){
  s <- sum(x)
  n <- length(x)
  results<- list(sum = s, length = n, answer = s / n)
  return(results)
}
```

```r
my_sum(1:10)
```

```
$sum
[1] 55

$length
[1] 10

$answer
[1] 5.5
```

```r
my_sum <- function(x){
  s <- sum(x)
  n <- length(x)
  results<- list(sum = s, length = n, answer = s / n)
  return(results)
}
```

```r
answer <- my_sum(1:10)
answer$answer
```

```
[1] 5.5
```

```r
answer[['answer']]
```

```
[1] 5.5
```

```r
names(answer)
```

```
[1] "sum"    "length" "answer"
```

```r
x <- 1:10
x[3] <- NA
my_sum(x)
```

```
$sum
[1] NA

$length
[1] 10

$answer
[1] NA
```

```r
my_sum <- function(x){
  s <- sum(x, na.rm=T)
  n <- length(!is.na(x))
  results <- list("sum" = s, "length" = n, "answer" = s/n)
}
my_sum(x)
```

```r
my_sum <- function(x){
  s <- sum(x, na.rm = T)
  n <- length(!is.na(x))
  results <- list("sum" = s, "length" = n, "answer" = s/n)
  return(results) #<<
}
my_sum(x)
```

```
$sum
[1] 52

$length
[1] 10

$answer
[1] 5.2
```

```r
my_sum <- function(x){
  s <- sum(x, na.rm = T)
  n <- length(!is.na(x))
  results <- list("sum" = s, "length" = n, "answer" = s/n)
  return(results)
}
```

```r
my_sum <- function(x){
  s <- sum(x, na.rm = T)
{{  n <- sum(!is.na(x)) }}
  results <- list("sum" = s, "length" = n, "answer" = s/n)
  return(results)
}
my_sum(x)
```

```
$sum
[1] 52

$length
[1] 9

$answer
[1] 5.777778
```

```
my_sum <- function(x){
  s <- sum(x, na.rm = T)
  n <- sum(!is.na(x))
  results <- list("sum" = s, "length" = n, "answer" = s/n)
  return(results)
}
```

```
my_sum <- function(x, remove_missing = TRUE){ #<<
  s <- sum(x, na.rm = T)
  n <- sum(!is.na(x))
  results <- list("sum" = s, "length" = n, "answer" = s/n)
  return(results)
}
```

```
my_sum <- function(x, remove_missing = TRUE){
  {{if(remove_missing){
    x <- x[!is.na(x)]
  }
  s <- sum(x)
  n <- length(x)}}
  results <- list("sum" = s, "length" = n, "answer" = s/n)
  return(results)
}
my_sum(x)
```

```
$sum
[1] 52

$length
[1] 9

$answer
[1] 5.777778
```

```
my_sum <- function(x, remove_missing = TRUE){
  if(remove_missing){
    x <- x[!is.na(x)]
  }
  s <- sum(x)
  n <- length(x)
  results <- list("sum" = s, "length" = n, "answer" = s/n, "nmiss" = sum(is.na(x)))
  return(results)
}
my_sum(x)
```

```
$sum
[1] 52
```

```
$length
[1] 9

$answer
[1] 5.777778

$nmiss
[1] 0
```

```
my_sum <- function(x, remove_missing = TRUE){
  nmiss <- sum(is.na(x)) #<<
  if(remove_missing){
    x <- x[!is.na(x)]
  }
  s <- sum(x)
  n <- length(x)
  results <- list("sum" = s, "length" = n, "answer" = s/n, "nmiss" = sum(is.na(x)))
  return(results)
}
my_sum(x)
```

```
$sum
[1] 52

$length
[1] 9

$answer
[1] 5.777778

$nmiss
[1] 0
```

```
my_sum <- function(x, remove_missing = TRUE){
  nmiss <- sum(is.na(x))
  if(remove_missing){
    x <- x[!is.na(x)]
  }
  s <- sum(x)
  n <- length(x)
  results <- list("sum" = s, "length" = n, "answer" = s/n, "nmiss" = nmiss) #<<
  return(results)
}
my_sum(x)
```

```
$sum
[1] 52

$length
```

```
[1] 9

$answer
[1] 5.777778

$nmiss
[1] 1
```

```
my_sum(x, remove_missing = F)
```

```
$sum
[1] NA

$length
[1] 10

$answer
[1] NA

$nmiss
[1] 1
```

```
my_summary <- function(d){

}
```

```
my_summary <- function(d){
  require(tidyverse) #<
}
```

```
my_summary <- function(d){
  require(tidyverse)
  summary_cts <- d %>%
    summarize_if(is.numeric, list("mean" = ~mean(x, na.rm=T),
                                  "median" = ~median(x, na.rm=T),
                                  'sd' = ~sd(x, na.rm=T),
                                  'nmiss' = ~sum(is.na(x))))
  return(list("cts" = summary_cts))
}
my_summary(iris)
```

```
Loading required package: tidyverse

-- Attaching packages ----------------------------- tidyverse 1.2.1 --

v ggplot2 3.1.0          v purrr   0.3.2
v tibble  2.0.1          v dplyr   0.8.0.9009
v tidyr   0.8.3          v stringr 1.4.0
v readr   1.3.1          v forcats 0.4.0

Warning: package 'tibble' was built under R version 3.5.2
```

```
Warning: package 'tidyr' was built under R version 3.5.2

Warning: package 'stringr' was built under R version 3.5.2

-- Conflicts -------------------------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()

$cts
  Sepal.Length_mean Sepal.Width_mean Petal.Length_mean Petal.Width_mean
1          5.777778         5.777778          5.777778         5.777778
  Sepal.Length_median Sepal.Width_median Petal.Length_median
1                   6                  6                   6
  Petal.Width_median Sepal.Length_sd Sepal.Width_sd Petal.Length_sd
1                  6        3.073181       3.073181        3.073181
  Petal.Width_sd Sepal.Length_nmiss Sepal.Width_nmiss Petal.Length_nmiss
1       3.073181                  1                 1                  1
  Petal.Width_nmiss
1                 1
```

```r
my_summary <- function(d){
  require(tidyverse)
  summary_cts <- d %>%
    summarize_if(is.numeric, list("mean" = ~mean(x, na.rm=T),
                                  "median" = ~median(x, na.rm=T),
                                  'sd' = ~sd(x, na.rm=T),
                                  'nmiss' = ~sum(is.na(x)))) %>%
    gather(variable, value) %>%
    separate(variable, c("variable","stat"), sep='_') %>%
    spread(stat, value)
  return(list("cts" = summary_cts))
}
my_summary(iris)
```

```
$cts
      variable     mean median nmiss       sd
1 Petal.Length 5.777778      6     1 3.073181
2  Petal.Width 5.777778      6     1 3.073181
3 Sepal.Length 5.777778      6     1 3.073181
4  Sepal.Width 5.777778      6     1 3.073181
```

```r
my_summary <- function(d){
  require(tidyverse)
  summary_cts <- d %>%
    summarize_if(is.numeric, list("mean" = ~mean(x, na.rm=T), #<<
                                  "median" = ~median(x, na.rm=T),#<<
                                  'sd' = ~sd(x, na.rm=T),#<<
                                  'nmiss' = ~sum(is.na(x)))) %>% #<<
    gather(variable, value) %>%
    separate(variable, c("variable","stat"), sep='_') %>%
```

```
    spread(stat, value)
  return(list("cts" = summary_cts))
}
my_summary(iris)
```

```
$cts
      variable     mean median nmiss       sd
1 Petal.Length 5.777778      6     1 3.073181
2  Petal.Width 5.777778      6     1 3.073181
3 Sepal.Length 5.777778      6     1 3.073181
4  Sepal.Width 5.777778      6     1 3.073181
```

```
my_summary <- function(d){
  require(tidyverse)
  summary_cts <- d %>%
    summarize_if(is.numeric, list("mean" = ~mean(., na.rm=T),#<<
                                  "median" = ~median(., na.rm=T),#<<
                                  'sd' = ~sd(., na.rm=T),#<<
                                  'nmiss' = ~sum(is.na(.)))) %>% #<<
    gather(variable, value) %>%
    separate(variable, c("variable","stat"), sep='_') %>%
    spread(stat, value)
  return(list("cts" = summary_cts))
}
my_summary(iris)
```

```
$cts
      variable     mean median nmiss        sd
1 Petal.Length 3.758000   4.35     0 1.7652982
2  Petal.Width 1.199333   1.30     0 0.7622377
3 Sepal.Length 5.843333   5.80     0 0.8280661
4  Sepal.Width 3.057333   3.00     0 0.4358663
```

```
my_summary <- function(d){
  require(tidyverse)
  summary_cts <- d %>%
    summarize_if(is.numeric, list("mean" = ~mean(., na.rm=T),
                                  "median" = ~median(., na.rm=T),
                                  'sd' = ~sd(., na.rm=T),
                                  'nmiss' = ~sum(is.na(.)))) %>%
    gather(variable, value) %>%
    separate(variable, c("variable","stat"), sep='_') %>%
    spread(stat, value) %>%
    select(variable, nmiss, everything()) #<<
  return(list("cts" = summary_cts))
}
my_summary(iris)
```

```
$cts
      variable nmiss     mean median        sd
1 Petal.Length     0 3.758000   4.35 1.7652982
2  Petal.Width     0 1.199333   1.30 0.7622377
3 Sepal.Length     0 5.843333   5.80 0.8280661
4  Sepal.Width     0 3.057333   3.00 0.4358663
```

```
my_summary <- function(d){
  require(tidyverse)
  summary_cts <- d %>%
    summarize_if(is.numeric, list("mean" = ~mean(., na.rm=T),
                              "median" = ~median(., na.rm=T),
                              'sd' = ~sd(., na.rm=T),
                              'nmiss' = ~sum(is.na(.)))) %>%
    gather(variable, value) %>%
    separate(variable, c("variable","stat"), sep='_') %>%
    spread(stat, value) %>%
    select(variable, nmiss, everything())
  summary_cat <- d %>% #<<
    summarise_if(is.factor, list('nmiss' = ~sum(is.na(.)),#<<
                             'ncat' = ~length(unique(.)),#<<
                             'categories' = ~paste(sort(unique(levels(.))), collaps
              )#<<
  return(list("cts" = summary_cts,
              "cat" = summary_cat))
}
my_summary(iris)
```

```
$cts
      variable nmiss     mean median        sd
1 Petal.Length     0 3.758000   4.35 1.7652982
2  Petal.Width     0 1.199333   1.30 0.7622377
3 Sepal.Length     0 5.843333   5.80 0.8280661
4  Sepal.Width     0 3.057333   3.00 0.4358663

$cat
  nmiss ncat                     categories
1     0    3 setosa, versicolor, virginica
```

```
 my_summary <- function(d){
   require(tidyverse)
   if(!is.data.frame(d)){#<<
     stop("Input must be a data.frame")#<<
   }#<<
   summary_cts <- d %>%
     summarize_if(is.numeric, list("mean" = ~mean(., na.rm=T),
                               "median" = ~median(., na.rm=T),
                               'sd' = ~sd(., na.rm=T),
```

```
                                    'nmiss' = ~sum(is.na(.)))) %>%
     gather(variable, value) %>%
     separate(variable, c("variable","stat"), sep='_') %>%
     spread(stat, value) %>%
     select(variable, nmiss, everything())
   summary_cat <- d %>%
     summarise_if(is.factor, list('nmiss' = ~sum(is.na(.)),
                                   'ncat' = ~length(unique(.)),
                                   'categories' = ~paste(sort(unique(levels(.))), collap
                  )
   return(list("cts" = summary_cts,
               "cat" = summary_cat))
 }
 my_summary(x)
```

```
datas <- list('cars' = mtcars, 'iris' = iris, 'diamonds'= diamonds)
map(datas, my_summary)
```

```
$cars
$cars$cts
   variable nmiss       mean  median          sd
1        am     0   0.406250   0.000   0.4989909
2      carb     0   2.812500   2.000   1.6152000
3       cyl     0   6.187500   6.000   1.7859216
4      disp     0 230.721875 196.300 123.9386938
5      drat     0   3.596563   3.695   0.5346787
6      gear     0   3.687500   4.000   0.7378041
7        hp     0 146.687500 123.000  68.5628685
8       mpg     0  20.090625  19.200   6.0269481
9      qsec     0  17.848750  17.710   1.7869432
10       vs     0   0.437500   0.000   0.5040161
11       wt     0   3.217250   3.325   0.9784574

$cars$cat
data frame with 0 columns and 1 row


$iris
$iris$cts
      variable nmiss     mean median        sd
1 Petal.Length     0 3.758000   4.35 1.7652982
2  Petal.Width     0 1.199333   1.30 0.7622377
3 Sepal.Length     0 5.843333   5.80 0.8280661
4  Sepal.Width     0 3.057333   3.00 0.4358663

$iris$cat
  nmiss ncat                    categories
```

```
1     0    3 setosa, versicolor, virginica


$diamonds
$diamonds$cts
# A tibble: 7 x 5
  variable nmiss     mean  median       sd
  <chr>    <dbl>    <dbl>   <dbl>    <dbl>
1 carat        0    0.798    0.7     0.474
2 depth        0   61.7     61.8     1.43
3 price        0 3933.    2401    3989.
4 table        0   57.5     57       2.23
5 x            0    5.73     5.7     1.12
6 y            0    5.73     5.71    1.14
7 z            0    3.54     3.53    0.706

$diamonds$cat
# A tibble: 1 x 9
  cut_nmiss color_nmiss clarity_nmiss cut_ncat color_ncat clarity_ncat
      <int>       <int>         <int>    <int>      <int>        <int>
1         0           0             0        5          7            8
# ... with 3 more variables: cut_categories <chr>, color_categories <chr>,
#   clarity_categories <chr>
```

Chapter 3

Modeling

```r
library(survival)
data(pbc)
str(pbc)
```

```
'data.frame':	418 obs. of  20 variables:
 $ id      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ time    : int  400 4500 1012 1925 1504 2503 1832 2466 2400 51 ...
 $ status  : int  2 0 2 2 1 2 0 2 2 2 ...
 $ trt     : int  1 1 1 1 2 2 2 2 1 2 ...
 $ age     : num  58.8 56.4 70.1 54.7 38.1 ...
 $ sex     : Factor w/ 2 levels "m","f": 2 2 1 2 2 2 2 2 2 2 ...
 $ ascites : int  1 0 0 0 0 0 0 0 0 1 ...
 $ hepato  : int  1 1 0 1 1 1 1 0 0 0 ...
 $ spiders : int  1 1 0 1 1 0 0 0 1 1 ...
 $ edema   : num  1 0 0.5 0.5 0 0 0 0 0 1 ...
 $ bili    : num  14.5 1.1 1.4 1.8 3.4 0.8 1 0.3 3.2 12.6 ...
 $ chol    : int  261 302 176 244 279 248 322 280 562 200 ...
 $ albumin : num  2.6 4.14 3.48 2.54 3.53 3.98 4.09 4 3.08 2.74 ...
 $ copper  : int  156 54 210 64 143 50 52 52 79 140 ...
 $ alk.phos: num  1718 7395 516 6122 671 ...
 $ ast     : num  137.9 113.5 96.1 60.6 113.2 ...
 $ trig    : int  172 88 55 92 72 63 213 189 88 143 ...
 $ platelet: int  190 221 151 183 136 NA 204 373 251 302 ...
 $ protime : num  12.2 10.6 12 10.3 10.9 11 9.7 11 11 11.5 ...
 $ stage   : int  4 3 4 4 3 3 3 3 2 4 ...
```

```r
myLinearModel <- lm(chol ~ bili, data = pbc)
```

```r
myLinearModel
```

```

Call:
lm(formula = chol ~ bili, data = pbc)
```

```
Coefficients:
(Intercept)         bili
     303.20        20.24
```

```r
summary(myLinearModel)
```

```
Call:
lm(formula = chol ~ bili, data = pbc)

Residuals:
    Min      1Q  Median      3Q     Max
-565.39  -89.90  -35.36   44.92 1285.33

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  303.204     15.601  19.435  < 2e-16 ***
bili          20.240      2.785   7.267 3.63e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 213.2 on 282 degrees of freedom
  (134 observations deleted due to missingness)
Multiple R-squared:  0.1577,    Adjusted R-squared:  0.1547
F-statistic:  52.8 on 1 and 282 DF,  p-value: 3.628e-12
```

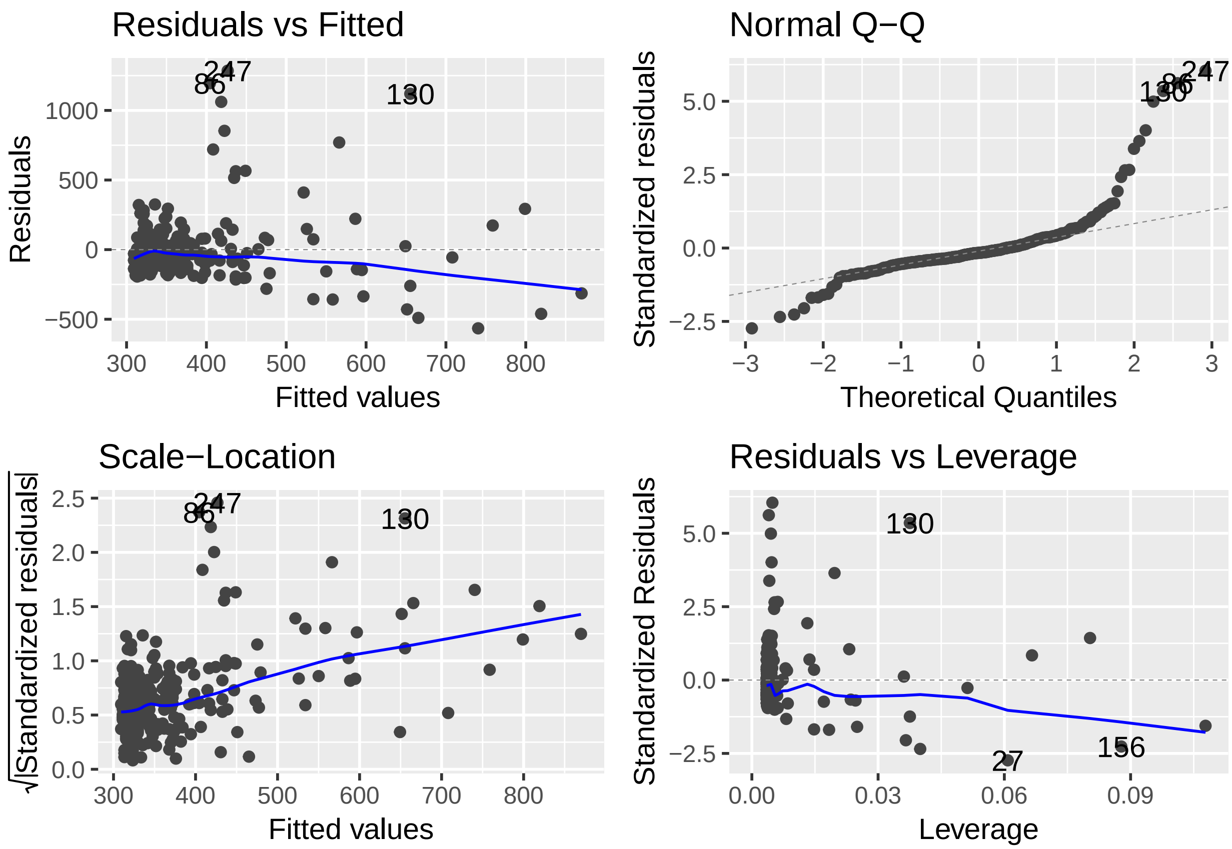
```r
broom::tidy(myLinearModel)
```

```
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    303.       15.6       19.4  5.65e-54
2 bili            20.2       2.79      7.27 3.63e-12
```

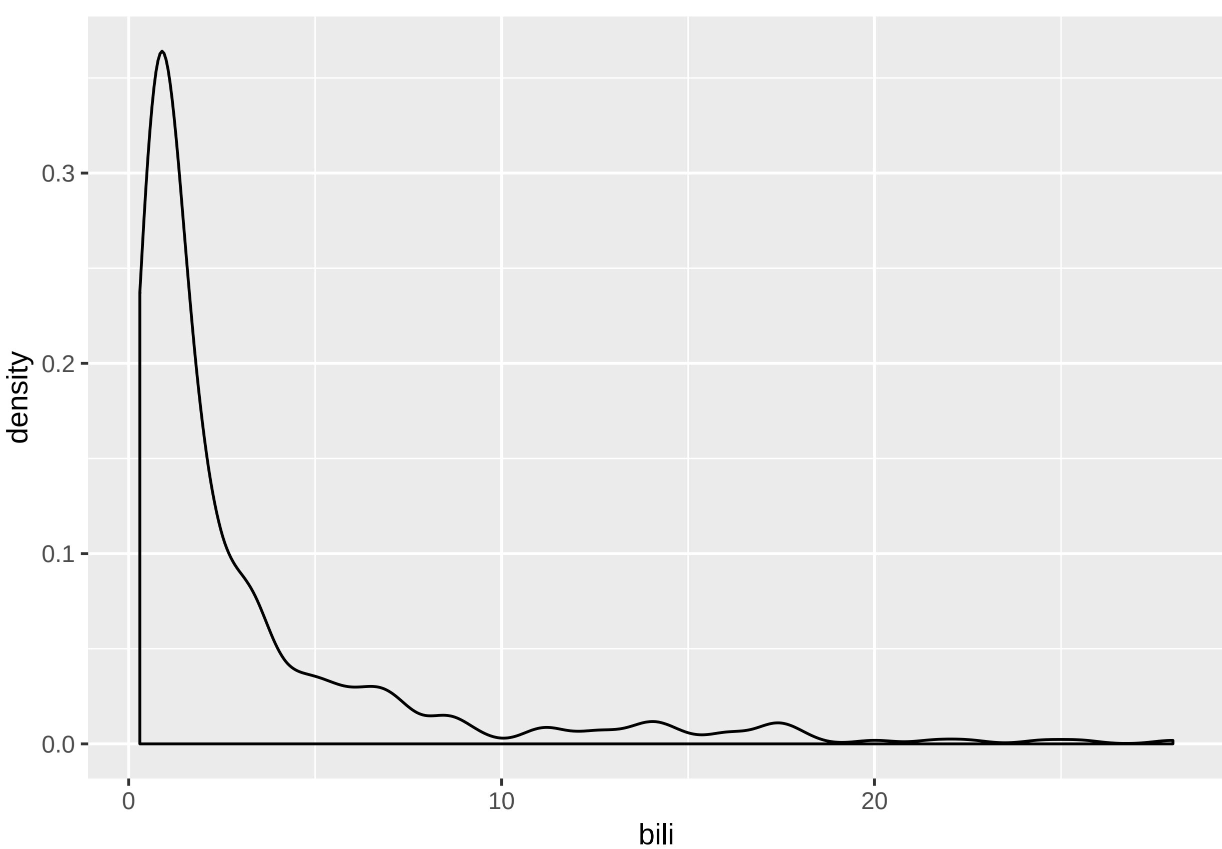
```r
broom::glance(myLinearModel)
```

```
# A tibble: 1 x 11
  r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC   BIC
      <dbl>         <dbl> <dbl>     <dbl>    <dbl> <int>  <dbl> <dbl> <dbl>
1     0.158         0.155  213.      52.8 3.63e-12     2 -1925. 3856. 3867.
# ... with 2 more variables: deviance <dbl>, df.residual <int>
```

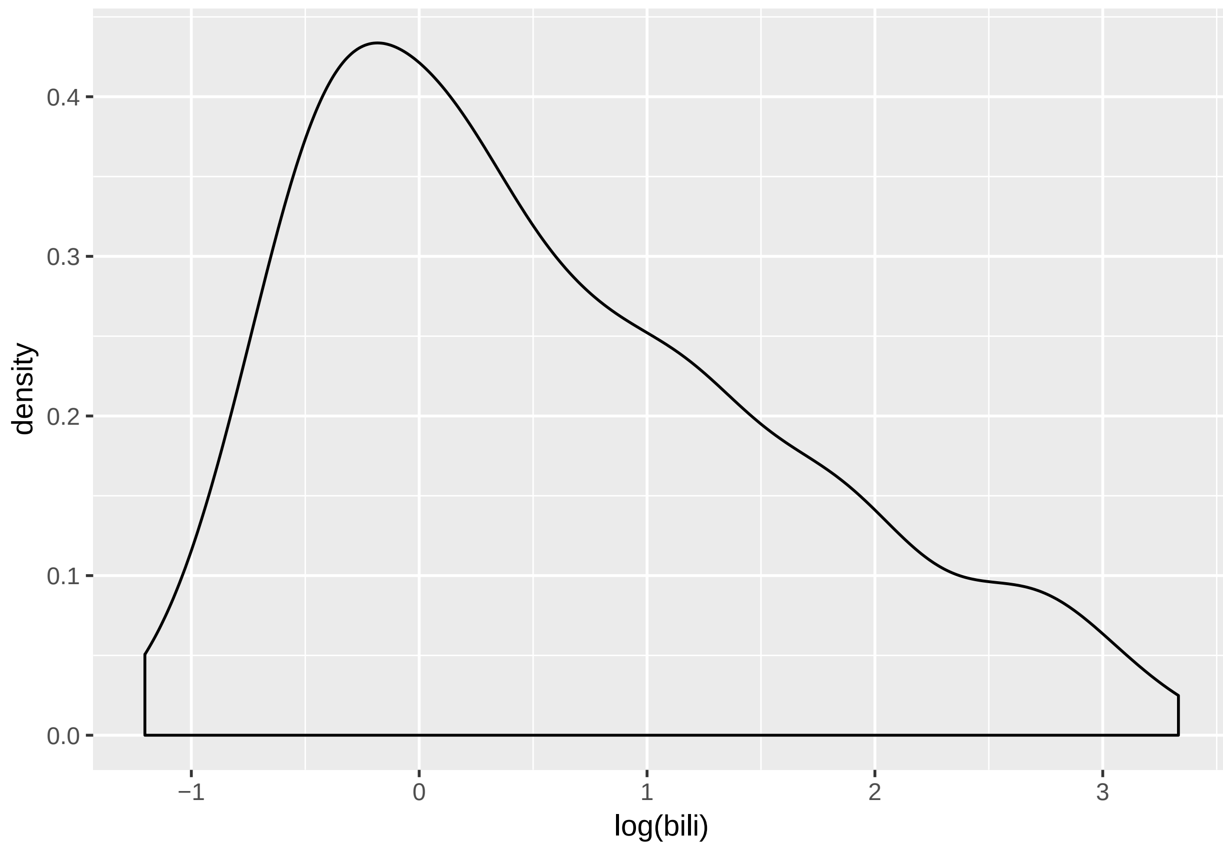
```r
## # install.packages('ggfortify')
## library(ggfortify)
## autoplot(myLinearModel)
```




```
## ggplot(pbc, aes(x = bili))+geom_density()
```



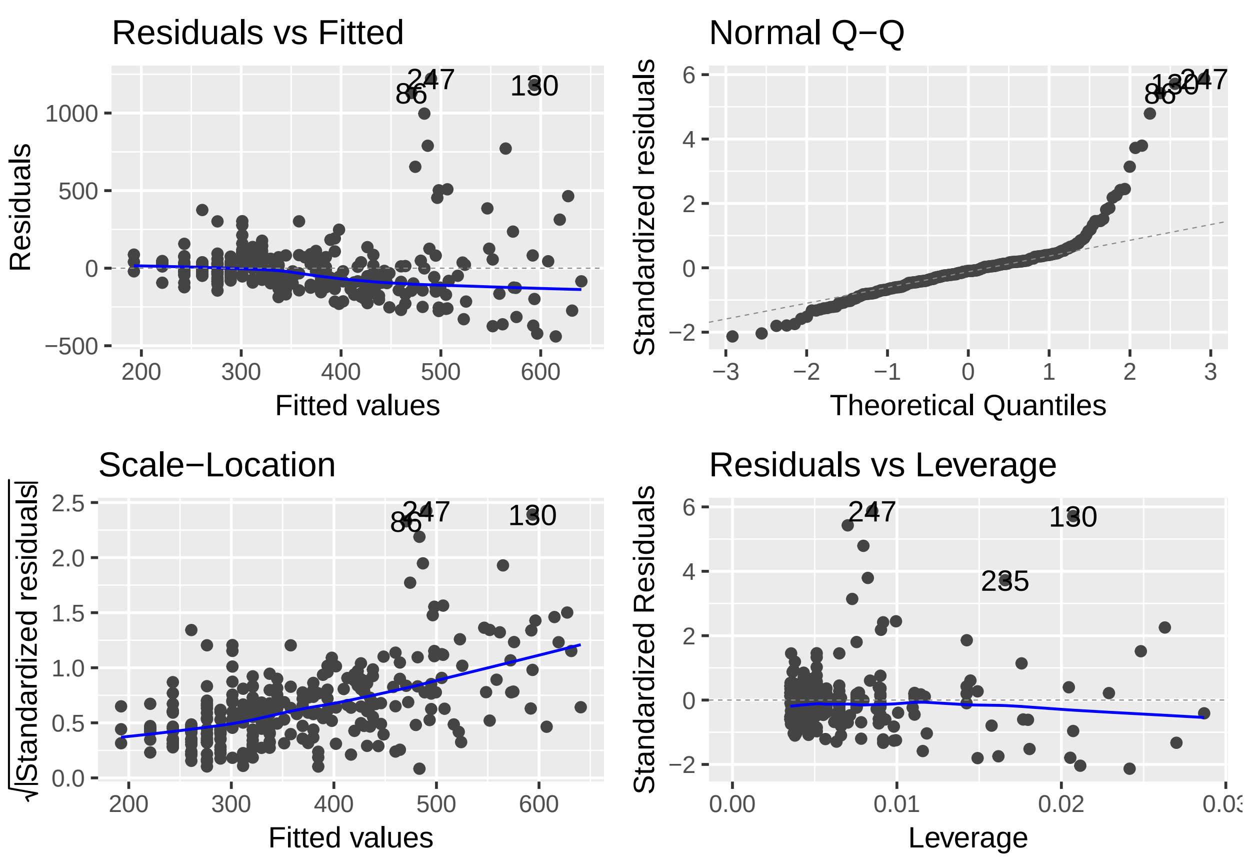
```
## ggplot(pbc, aes(x = log(bili)))+geom_density()
```



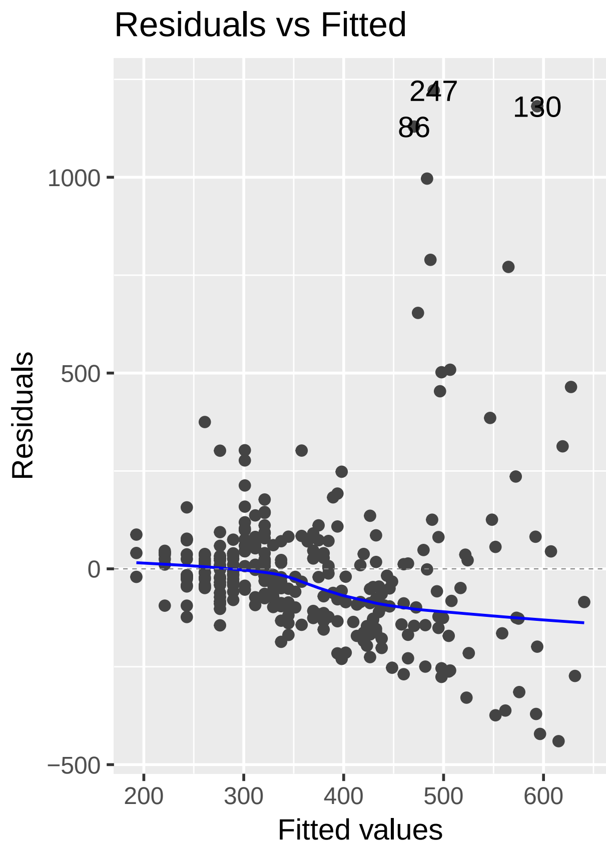

```
myLinearModel2 <- lm(chol~log(bili), data = pbc)
summary(myLinearModel2)
```

```
Call:
lm(formula = chol ~ log(bili), data = pbc)

Residuals:
    Min      1Q  Median      3Q     Max 
-440.07  -94.35  -21.07   42.67 1221.86 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   311.48      14.28  21.816  < 2e-16 ***
log(bili)      98.80      12.07   8.186 9.42e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 208.9 on 282 degrees of freedom
  (134 observations deleted due to missingness)
Multiple R-squared:  0.192,	Adjusted R-squared:  0.1891 
F-statistic: 67.01 on 1 and 282 DF,  p-value: 9.416e-15
```

```r
autoplot(myLinearModel2)
```



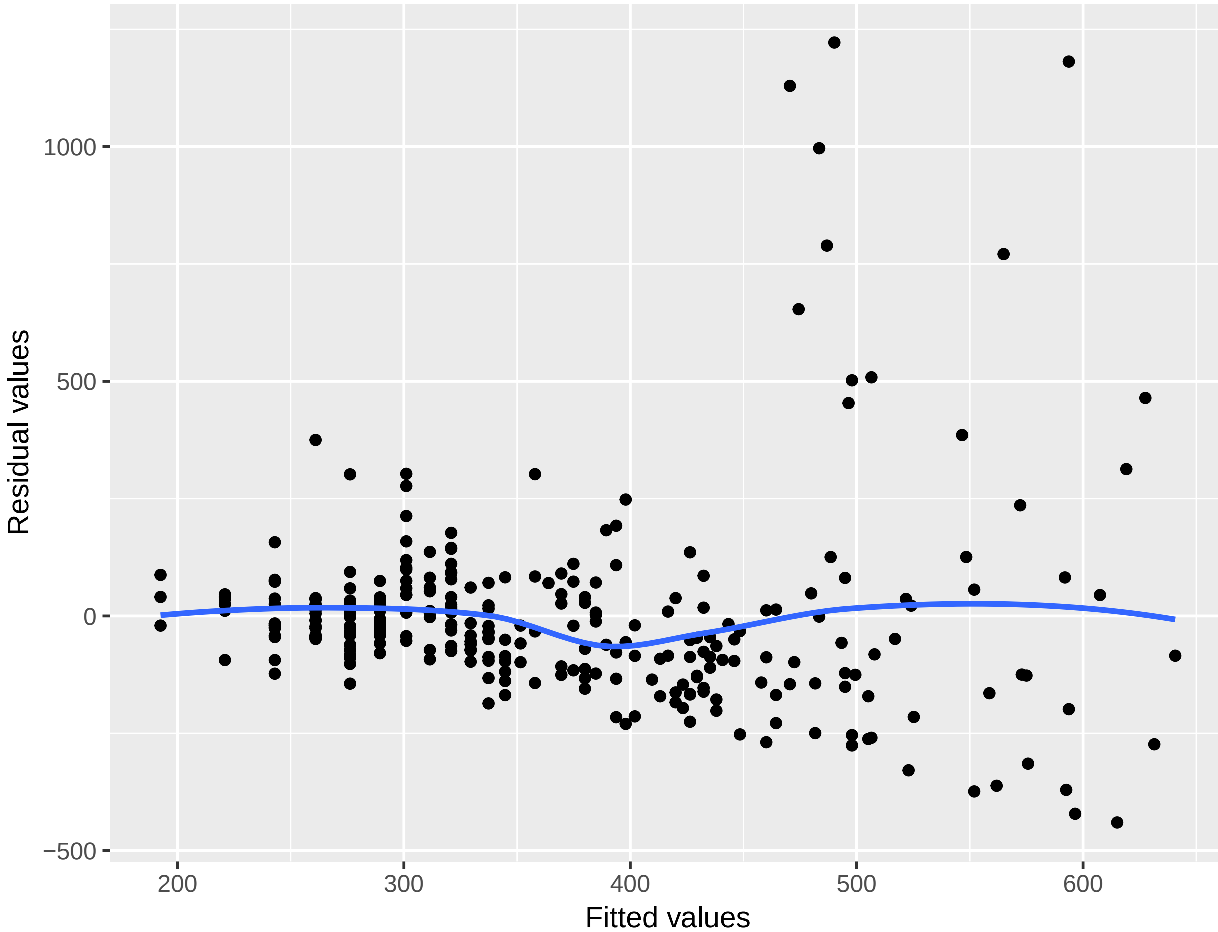

```r
autoplot(myLinearModel2, which=1)
```




```
d <- broom::augment(myLinearModel2)
d
```

```
# A tibble: 284 x 10
   .rownames  chol log.bili. .fitted .se.fit .resid    .hat .sigma .cooksd
   <chr>     <int>     <dbl>   <dbl>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>
 1 1           261    2.67     576.     28.1 -315.  0.0181    208. 2.13e-2
 2 2           302    0.0953   321.     13.7  -18.9 0.00433   209. 1.79e-5
 3 3           176    0.336    345.     12.8 -169.  0.00373   209. 1.23e-3
 4 4           244    0.588    370.     12.4 -126.  0.00352   209. 6.41e-4
 5 5           279    1.22     432.     14.6 -153.  0.00487   209. 1.33e-3
 6 6           248   -0.223    289.     15.8  -41.4 0.00571   209. 1.14e-4
 7 7           322    0        311.     14.3   10.5 0.00467   209. 5.98e-6
 8 8           280   -1.20     193.     24.9   87.5 0.0142    209. 1.28e-3
 9 9           562    1.16     426.     14.2  136.  0.00463   209. 9.84e-4
10 10          200    2.53     562.     26.6 -362.  0.0162    208. 2.51e-2
# ... with 274 more rows, and 1 more variable: .std.resid <dbl>
```

```
ggplot(d, aes(x = .fitted, y = .resid))+geom_point()+ geom_smooth(se=F)+
  labs(x = 'Fitted values', y = 'Residual values')
```




```
head(predict(myLinearModel2, newdata = pbc))
```

```
       1        2        3        4        5        6 
575.6925 320.9006 344.7277 369.5578 432.3941 289.4371 
```

```
myLM3 <- lm(chol ~ log(bili) + sex, data = pbc)
broom::tidy(myLM3)
```

```
# A tibble: 3 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)    283.       36.6     7.71  2.14e-13
2 log(bili)       99.6      12.1     8.22  7.37e-15
3 sexf            32.5      37.8     0.858 3.92e- 1
```

```
myLR <- glm(spiders ~ albumin + bili + chol, data = pbc, family = binomial)
myLR
```

```

Call:  glm(formula = spiders ~ albumin + bili + chol, family = binomial, 
    data = pbc)

Coefficients:
(Intercept)      albumin         bili         chol  
```

```
  2.3326484   -0.9954927    0.0995915   -0.0003176

Degrees of Freedom: 283 Total (i.e. Null);  280 Residual
  (134 observations deleted due to missingness)
Null Deviance:      341.4
Residual Deviance: 315.2    AIC: 323.2
```

```
broom::tidy(myLR)
```

```
# A tibble: 4 x 5
  term         estimate std.error statistic p.value
  <chr>           <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)  2.33      1.30          1.80  0.0717
2 albumin     -0.995     0.362        -2.75  0.00595
3 bili         0.0996    0.0344        2.89  0.00381
4 chol        -0.000318  0.000615     -0.517 0.605
```

```
broom::glance(myLR)
```

```
# A tibble: 1 x 7
  null.deviance df.null logLik   AIC   BIC deviance df.residual
          <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>       <int>
1          341.     283  -158.  323.  338.     315.          280
```

```
head(predict(myLR))
```

```
          1           2           3           4           5           6
 1.10554163 -1.77506554 -1.04814132 -0.09414055 -0.93144911 -1.62851203
```

```
head(predict(myLR, type='response'))
```

```
        1         2         3         4         5         6
0.7512970 0.1449135 0.2595822 0.4764822 0.2826308 0.1640343
```

3.1 Model selection

```
# install.packages('leaps')
library(leaps)
mtcars1 <- mtcars %>% mutate_at(vars(cyl, vs:carb), as.factor)
all_subsets <- regsubsets(mpg~., data = mtcars1)
all_subsets
```

```
Subset selection object
Call: regsubsets.formula(mpg ~ ., data = mtcars1)
16 Variables  (and intercept)
      Forced in Forced out
cyl6      FALSE      FALSE
cyl8      FALSE      FALSE
```

```
disp        FALSE       FALSE
hp          FALSE       FALSE
drat        FALSE       FALSE
wt          FALSE       FALSE
qsec        FALSE       FALSE
vs1         FALSE       FALSE
am1         FALSE       FALSE
gear4       FALSE       FALSE
gear5       FALSE       FALSE
carb2       FALSE       FALSE
carb3       FALSE       FALSE
carb4       FALSE       FALSE
carb6       FALSE       FALSE
carb8       FALSE       FALSE
1 subsets of each size up to 8
Selection Algorithm: exhaustive
```

```
ind <- which.max(summary(all_subsets)$adjr2)
summary(all_subsets)$which[ind,]
```

```
(Intercept)        cyl6        cyl8        disp          hp        drat
       TRUE        TRUE       FALSE       FALSE        TRUE       FALSE
         wt        qsec         vs1         am1       gear4       gear5
       TRUE       FALSE        TRUE        TRUE       FALSE       FALSE
      carb2       carb3       carb4       carb6       carb8
      FALSE       FALSE       FALSE       FALSE       FALSE
```

3.2 Many models

```
mtcars <- as_tibble(mtcars)
mtcars %>% select(mpg, disp:qsec)
```

```
# A tibble: 32 x 6
     mpg  disp    hp  drat    wt  qsec
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1  21    160    110  3.9   2.62  16.5
 2  21    160    110  3.9   2.88  17.0
 3  22.8  108     93  3.85  2.32  18.6
 4  21.4  258    110  3.08  3.22  19.4
 5  18.7  360    175  3.15  3.44  17.0
 6  18.1  225    105  2.76  3.46  20.2
 7  14.3  360    245  3.21  3.57  15.8
 8  24.4  147.    62  3.69  3.19  20
 9  22.8  141.    95  3.92  3.15  22.9
10  19.2  168.   123  3.92  3.44  18.3
# ... with 22 more rows
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg)
```

```
# A tibble: 160 x 3
     mpg variable value
   <dbl> <chr>    <dbl>
 1  21   disp      160
 2  21   disp      160
 3  22.8 disp      108
 4  21.4 disp      258
 5  18.7 disp      360
 6  18.1 disp      225
 7  14.3 disp      360
 8  24.4 disp      147.
 9  22.8 disp      141.
10  19.2 disp      168.
# ... with 150 more rows
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  group_by(variable) %>%
  lm(mpg~value, data=.)
```

```
Call:
lm(formula = mpg ~ value, data = .)

Coefficients:
(Intercept)        value
   21.28328     -0.01483
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable)
```

```
# A tibble: 5 x 2
  variable data
  <chr>    <list>
1 disp     <tibble [32 x 2]>
2 hp       <tibble [32 x 2]>
3 drat     <tibble [32 x 2]>
4 wt       <tibble [32 x 2]>
5 qsec     <tibble [32 x 2]>
```

```
bl <- mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable)
bl$data[[1]]
```

```
# A tibble: 32 x 2
     mpg value
   <dbl> <dbl>
 1  21    160
 2  21    160
 3  22.8  108
 4  21.4  258
 5  18.7  360
 6  18.1  225
 7  14.3  360
 8  24.4  147.
 9  22.8  141.
10  19.2  168.
# ... with 22 more rows
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable) %>%
  mutate(models = map(data, ~lm(mpg~value, data=.)))
```

```
# A tibble: 5 x 3
  variable data              models
  <chr>    <list>            <list>
1 disp     <tibble [32 x 2]> <S3: lm>
2 hp       <tibble [32 x 2]> <S3: lm>
3 drat     <tibble [32 x 2]> <S3: lm>
4 wt       <tibble [32 x 2]> <S3: lm>
5 qsec     <tibble [32 x 2]> <S3: lm>
```

```
 mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable) %>%
  mutate(models = map(data, ~lm(mpg~value, data=.)),
         outputs = map(models, ~tidy(.)))
```

```
# A tibble: 5 x 4
  variable data              models   outputs
  <chr>    <list>            <list>   <list>
1 disp     <tibble [32 x 2]> <S3: lm> <tibble [2 x 5]>
2 hp       <tibble [32 x 2]> <S3: lm> <tibble [2 x 5]>
3 drat     <tibble [32 x 2]> <S3: lm> <tibble [2 x 5]>
4 wt       <tibble [32 x 2]> <S3: lm> <tibble [2 x 5]>
5 qsec     <tibble [32 x 2]> <S3: lm> <tibble [2 x 5]>
```

```
 mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable) %>%
  mutate(models = map(data, ~lm(mpg~value, data=.)),
         outputs = map(models, ~tidy(.))) %>%
```

```
  select(variable, outputs)
```

```
# A tibble: 5 x 2
  variable outputs
  <chr>    <list>
1 disp     <tibble [2 x 5]>
2 hp       <tibble [2 x 5]>
3 drat     <tibble [2 x 5]>
4 wt       <tibble [2 x 5]>
5 qsec     <tibble [2 x 5]>
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable) %>%
  mutate(models = map(data, ~lm(mpg~value, data=.)),
         outputs = map(models, ~tidy(.))) %>%
  select(variable, outputs) %>%
  unnest()
```

```
# A tibble: 10 x 6
   variable term        estimate std.error statistic  p.value
   <chr>    <chr>          <dbl>     <dbl>     <dbl>    <dbl>
 1 disp     (Intercept)  29.6       1.23        24.1   3.58e-21
 2 disp     value        -0.0412    0.00471     -8.75  9.38e-10
 3 hp       (Intercept)  30.1       1.63        18.4   6.64e-18
 4 hp       value        -0.0682    0.0101      -6.74  1.79e- 7
 5 drat     (Intercept)  -7.52      5.48        -1.37  1.80e- 1
 6 drat     value         7.68      1.51         5.10  1.78e- 5
 7 wt       (Intercept)  37.3       1.88        19.9   8.24e-19
 8 wt       value        -5.34      0.559       -9.56  1.29e-10
 9 qsec     (Intercept)  -5.11     10.0         -0.510 6.14e- 1
10 qsec     value         1.41      0.559        2.53  1.71e- 2
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable) %>%
  mutate(models = map(data, ~lm(mpg~value, data=.)),
         outputs = map(models, ~tidy(.))) %>%
  select(variable, outputs) %>%
  unnest() %>%
  filter(term=='value')
```

```
# A tibble: 5 x 6
  variable term  estimate std.error statistic  p.value
  <chr>    <chr>    <dbl>     <dbl>     <dbl>    <dbl>
1 disp     value  -0.0412   0.00471     -8.75 9.38e-10
2 hp       value  -0.0682   0.0101      -6.74 1.79e- 7
3 drat     value   7.68     1.51         5.10 1.78e- 5
```

```
4 wt        value   -5.34     0.559       -9.56 1.29e-10
5 qsec      value    1.41     0.559        2.53 1.71e- 2
```

```
mtcars %>% select(mpg, disp:qsec) %>%
  gather(variable, value, -mpg) %>%
  nest(-variable) %>%
  mutate(models = map(data, ~lm(mpg~value, data=.)),
         outputs = map(models, ~tidy(.))) %>%
  select(variable, outputs) %>%
  unnest() %>%
  filter(term=='value') %>%
  mutate_if(is.numeric, funs(round(., 3)))
```

```
# A tibble: 5 x 6
  variable term  estimate std.error statistic p.value
  <chr>    <chr>    <dbl>     <dbl>     <dbl>   <dbl>
1 disp     value   -0.041     0.005     -8.75   0
2 hp       value   -0.068     0.01      -6.74   0
3 drat     value    7.68      1.51       5.10   0
4 wt       value   -5.34      0.559     -9.56   0
5 qsec     value    1.41      0.559      2.52   0.017
```

Chapter 4

Predictive modeling

```
library(tidyverse)
library(caret)
data(diamonds)
set.seed(12356)
diamonds_train <- diamonds %>% sample_frac(size = 0.8) # 80%
diamonds_test <- anti_join(diamonds, diamonds_train)
(nrow(diamonds) == nrow(diamonds_train) + nrow(diamonds_test))
```

```
[1] FALSE
```