1.4 Bootstrap CIs

We will look at five different ways to create confidence intervals using the boostrap and discuss which to use when.

1. Percentile Bootstrap CI
2. Basic Bootstrap CI
3. Standard Normal Bootstrap CI
4. Bootstrap t (studentized)
5. Accelerated Bias-Corrected (BCa)

   " adjusted for skewness

Also which method to use when!

Key ideas:

1. When you say "we used bootstrapping to estimate CI" you need to say which one.

2. Whatever you are bootstrapping needs to be independent.

3. Bootstrapping is an attempt to simulate replication.
   (think about interpretation of a CI)

1.4.1 Percentile Bootstrap CI

Let $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$ be bootstrap replicates and let $\hat{\theta}_{\alpha/2}$ be the $\alpha/2$ quantile of $\underbrace{\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}}_{\text{bootstrap distribution.}}$.

Then, the $100(1-\alpha)\%$ Percentile Bootstrap CI for θ is

$$\left(\hat{\theta}_{\alpha/2} , \hat{\theta}_{1-\alpha/2} \right)$$

In R, if `bootstrap.reps = c(`$\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$`)`, the percentile CI is

vector of bootstrap statistics.

```
quantile(bootstrap.reps, c(alpha/2, 1 - alpha/2))
```

Assumptions/usage

1. Widely used because simple to implement & explain.
2. Use when little bias and skewness in bootstrap distribution.
3. Drawback: CI's can be too narrow (coverage will be low).
4. BCa intervals usually perform better (nominal coverage).

1.4.2 Basic Bootstrap CI (Corrects for bias)

The $100(1-\alpha)\%$ Basic Bootstrap CI for θ is

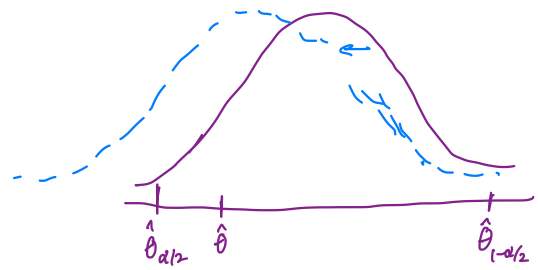
$$\left(\hat{\theta} - [\hat{\theta}_{1-\alpha/2} - \hat{\theta}],\ \hat{\theta} - [\hat{\theta}_{\alpha/2} - \hat{\theta}]\right)$$

Look at dsn of $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$

estimate from original data sample

quantiles from bootstrap distribution

$$\Rightarrow \left(2\hat{\theta} - \hat{\theta}_{1-\alpha/2},\ 2\hat{\theta} - \hat{\theta}_{\alpha/2}\right)$$



Assumptions/usage

1. Better than percentile bootstrap (when bootstrap dsn is biased) because corrects for bias. (does nothing for skewness).

2. Harder to explain.

1.4.3 Standard Normal Bootstrap CI (least favorite).

From the CLT,

$$Z = \frac{\hat{\theta} - E(\hat{\theta})}{se(\hat{\theta})} \overset{\circ}{\sim} N(0,1).$$

So, the $100(1-\alpha)\%$ Standard Normal Bootstrap CI for θ is

$$\hat{\theta} \pm z_{1-\alpha/2}\, \hat{se}(\hat{\theta}).$$

estimated $se(\hat{\theta})$ comes from bootstrap.

$sd(\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)})$.

Assumptions/usage

1. $\hat{\theta} \overset{\circ}{\sim} N\left(E(\hat{\theta}),\ se(\hat{\theta})^2\right)$

   This is a big assumption if $\hat{\theta}$ is not a sample mean!

2. $\hat{\theta}$ is unbiased $\Rightarrow$ $E(\hat{\theta}) = \theta$

   (can use bias correction w/ this method also)

   ↳ see code later

3. typically requires large n.

misleading name

1.4.4 Bootstrap t CI (Studentized Bootstrap)

Even if the distribution of $\hat{\theta}$ is Normal and $\hat{\theta}$ is unbiased for θ, the Normal distribution is not exactly correct for z. (because we estimate $se(\hat{\theta})$).

$$t^* = \frac{\hat{\theta} - E(\hat{\theta})}{\hat{se}(\hat{\theta})} \sim t_{n-1} \, ? \; \times$$

Additionally, the distribution of $\hat{se}(\hat{\theta})$ is unknown.

So we can't claim $z^* \sim t_{n-1}$

⇒ The bootstrap t interval does not use a Student t distribution as the reference distribuion, instead we estimate the distribution of a "t type" statistic by resampling.

The $100(1-\alpha)\%$ Bootstrap t CI is

$1-\alpha/2$ quantile of the bootstrap "t-type" statistic.

$$\left(\hat{\theta} - t^*_{1-\alpha/2} \cdot \hat{se}(\hat{\theta}),\; \hat{\theta} + t^*_{\alpha/2}\, \hat{se}(\hat{\theta}) \right)$$

Overview

t-type statistic: $t^{(1)} = \dfrac{\hat{\theta}^{(1)} - \hat{\theta}}{\hat{se}(\hat{\theta}^{(1)})}, \ldots, t^{(B)} = \dfrac{\hat{\theta}^{(B)} - \hat{\theta}}{\hat{se}(\hat{\theta}^{(B)})}$

bootstrap estimate of se of 1st bootstrap statistic $\hat{\theta}^{(1)}$.

To estimate the "t style distribution" for $\hat{\theta}$,

DOUBLE BOOTSTRAP!! (woah)

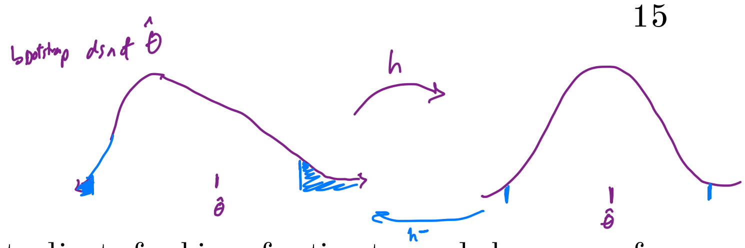
1. Compute $\hat{\theta}$
2. For each replicate $b = 1, \ldots, B$
   a) Sample w/ replacement from x
      $x^{(b)} = (x_1^{(b)}, \ldots, x_n^{(b)})$
   b) complete $\hat{\theta}^{(b)} = T(x^{(b)})$.
   c) For each replicate $r = 1, \ldots, R$
      i) sample w/ replacement from $x^{(b)}$
         $x^{(b)(r)} = (x_1^{(b)(r)}, \ldots, x_n^{(b)(r)})$
      ii) compute $\hat{\theta}^{(b)(r)} = T(x^{(b)(r)})$
   d) Compute $\hat{se}(\hat{\theta}^{(b)}) = sd(\hat{\theta}^{(b)(1)}, \ldots, \hat{\theta}^{(b)(R)})$
   e) Compute t-style statistic $t^{(b)} = \dfrac{\hat{\theta}^{(b)} - \hat{\theta}}{\hat{se}(\hat{\theta}^{(b)})}$
3. get quantiles.
   $t^*_{1-\alpha/2}, \; t^*_{\alpha/2}$
4. Compute CI

Assumptions/usage

1. Requires small bias and skewness in bootstrap dsn.

* 2. Computationally intensive.

3. Need $\hat{\theta}$ independedent of $\hat{se}(\hat{\theta})$.



"Accelerated Bias Corrected"

1.4.5 BCa CIs

Modified version of percentile intervals that adjusts for bias of estimator and skewness of the sampling distribution.

This method automatically selects a transformation so that the normality assumption holds.

Idea:

Assume there exists a monotonically increasing function g and constants a & b s.t.

$$\frac{g(\hat{\theta}) - g(\theta)}{1 + a\,g(\theta)} + b \sim N(0,1).$$

where $1 + a\,g(\theta) > 0$.

The BCa method uses bootstrapping to estimate the bias and skewness then modifies which percentiles are chosen to get the appropriate confidence limits for a given data set.

In summary,

BCa is like the percentile bootstrap but instead of $(\hat{\theta}_{\alpha/2}, \hat{\theta}_{1-\alpha/2})$,

BCa chooses better quantiles (not $\alpha/2$ & $1-\alpha/2$) to account for bias and skewness.

Assumptions/ Usage

1. Better performance than percentile bootstrap. (better coverage).

2. Harder to explain.

Your Turn

We will consider a telephone repair example from Hesterberg (2014). `Verizon` has repair times, with two groups, CLEC and ILEC, customers of the "Competitive" and "Incumbent" local exchange carrier.

other carriers

verizon customers.

Verizon required by law to serve both at the same speed.

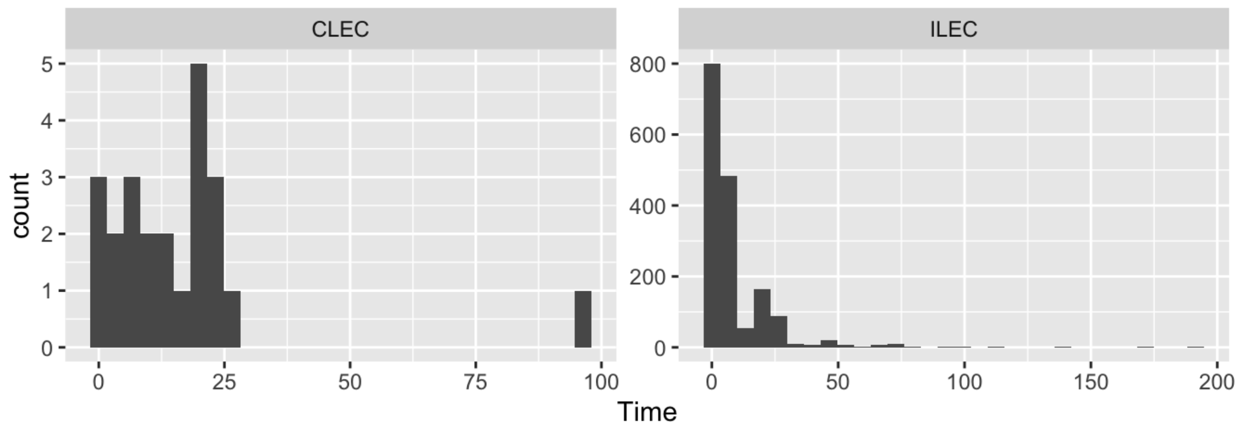
```
library(resample) # package containing the data

data(Verizon)
head(Verizon)
```

```
##    Time Group
## 1 17.50  ILEC
## 2  2.40  ILEC
## 3  0.00  ILEC
## 4  0.65  ILEC
## 5 22.23  ILEC
## 6  1.20  ILEC
```

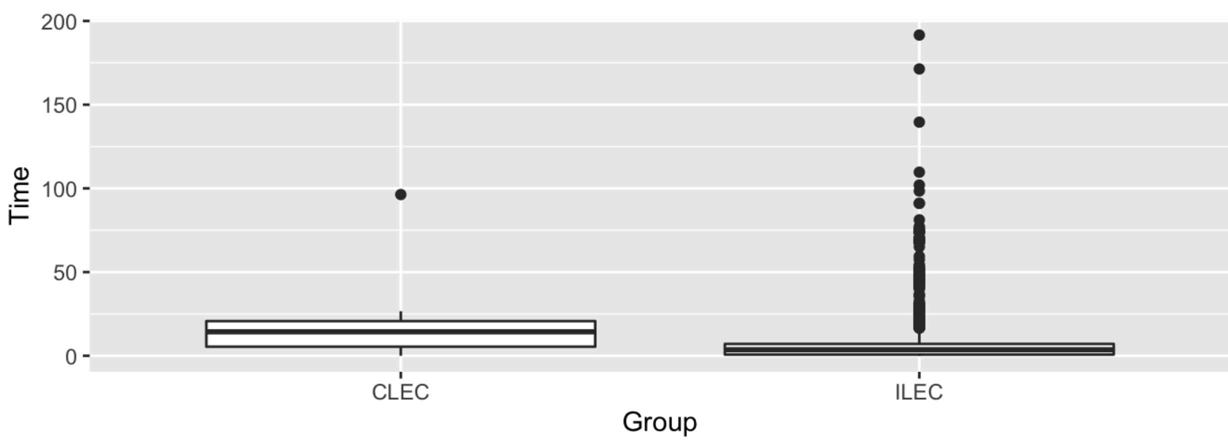
```
Verizon %>%
  group_by(Group) %>%
  summarize(mean = mean(Time), sd = sd(Time), min = min(Time), max =
 max(Time)) %>%
  kable()
```

	Group	mean	sd	min	max	n
other company customers	CLEC	16.509130	19.50358	0	96.32	23
verizon customers.	ILEC	8.411611	14.69004	0	191.60	1664

```
ggplot(Verizon) +
  geom_histogram(aes(Time)) +
  facet_wrap(.~Group, scales = "free")
```




```r
ggplot(Verizon) +
  geom_boxplot(aes(Group, Time))
```



1.5 Bootstrapping CIs

There are many bootstrapping packages in `R`, we will use the `boot` package. The function `boot` generates R resamples of the data and computes the desired statistic(s) for each sample. This function requires 3 arguments:

1. data = the data from the original sample (data.frame or matrix).
2. statistic = a function to compute the statistic from the data where the first argument is the data and the second argument is the indices of the obervations in the boostrap sample.
3. R = the number of bootstrap replicates.

```r
library(boot) # package containing the bootstrap function
```

data vector) bootstrap indices.

```r
mean_func <- function(x, idx) {
  mean(x[idx])
}
```

resample.

just verizon customers.

```r
ilec_times <- Verizon[Verizon$Group == "ILEC",]$Time
boot.ilec <- boot(ilec_times, mean_func, 2000)
```

bootstrap samples $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}$

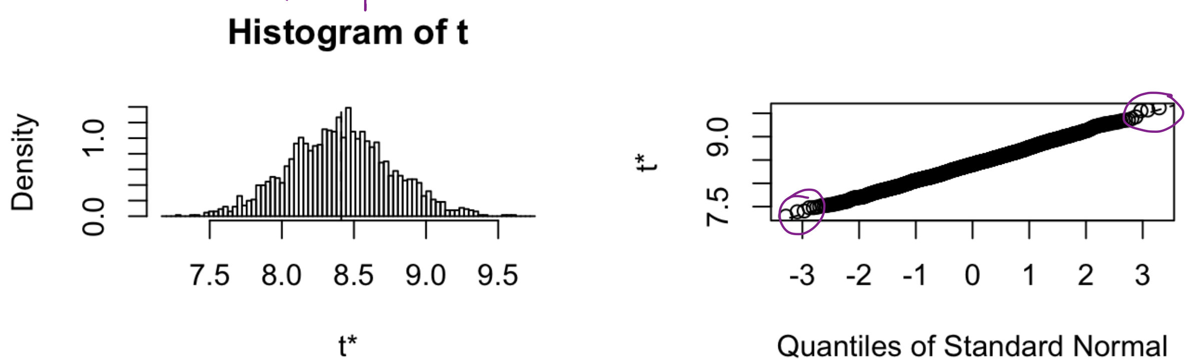
data vector

my function $\hat{\theta} = T(x)$

B.

```r
plot(boot.ilec)
```

Bootstrap dsn.



If we want to get Bootstrap CIs, we can use the **boot.ci** function to generate the 5 different nonparamteric bootstrap confidence intervals.

CI

```r
boot.ci(boot.ilec, conf = .95, type = c("perc", "basic", "norm",
  "bca"))
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.ilec, conf = 0.95, type = c("perc", "basic",
```

```
##     "norm", "bca"))
##
## Intervals :
## Level      Normal              Basic
## 95%   ( 7.719,  9.114 )   ( 7.709,  9.119 )
##
## Level     Percentile            BCa
## 95%   ( 7.704,  9.114 )   ( 7.752,  9.164 )
## Calculations and Intervals on Original Scale
```

bias-corrected.

```
## we can do some of these on our own
## normal
mean(boot.ilec$t) + c(-1, 1)*qnorm(.975)*sd(boot.ilec$t)
```

```
## [1] 7.709670 9.104182
```

$\hat{\theta}$

```
## normal is bias corrected
2*mean(ilec_times) - (mean(boot.ilec$t) - c(-1,
  1)*qnorm(.975)*sd(boot.ilec$t))
```

```
## [1] 7.719039 9.113551
```

```
## percentile
quantile(boot.ilec$t, c(.025, .975))
```

```
##     2.5%    97.5%
## 7.707656 9.111150
```

very similar => not much bias.

```
## basic
2*mean(ilec_times) - quantile(boot.ilec$t, c(.975, .025))
```

```
##    97.5%     2.5%
## 7.712071 9.115565
```

To get the studentized bootstrap CI, we need our statistic function to also return the variance of $\hat{\theta}$.

$\text{var}(\bar{X}) = \frac{\text{var} X}{n}$

```
mean_var_func <- function(x, idx) {
  c(mean(x[idx]), var(x[idx])/length(idx))
}

boot.ilec_2 <- boot(ilec_times, mean_var_func, 2000)
boot.ci(boot.ilec_2, conf = .95, type = "stud")
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 2000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot.ilec_2, conf = 0.95, type = "stud")
##
## Intervals :
## Level    Studentized
## 95%   ( 7.733,  9.231 )
## Calculations and Intervals on Original Scale
```

Which CI should we use?

All very similar, don't look skewed or biased.

Percentile ok in this case.

BCa good default choice (especially if not explaining it).

n large => Normal not a bad choice

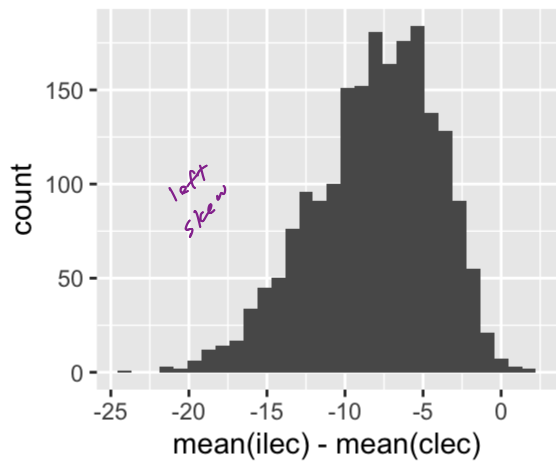
1.6 Bootstrapping for the difference of two means

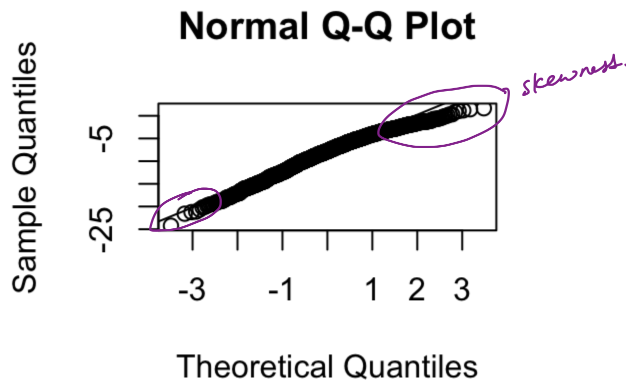
Given iid draws of size n and m from two populations, to compare the means of the two groups using the bootstrap,

1. For replicates $b = 1, \dots, B$
   a) Draw a resample of size n w/ replacement from sample 1 and separately of size m from sample 2.
   b) Compute a statistic that compares the two groups (i.e. $\hat{\theta} = \bar{x}_1 - \bar{x}_2$).
2. Construct a bootstrap dsn of statistic $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$ ← inspect shape, $\hat{\text{bias}}$, $\hat{se}$
3. Compute appropriate CI based on 2.

The function `two.boot` in the `simpleboot` package is used to bootstrap the difference between univariate statistics. Use the bootstrap to compute the shape, bias, and bootstrap sample error for the samples from the `Verizon` data set of CLEC and ILEC customers.

```
library(simpleboot)

clec_times <- Verizon[Verizon$Group == "CLEC",]$Time

diff_means.boot <- two.boot(ilec_times, clec_times, "mean", R = 2000)

ggplot() +
  geom_histogram(aes(diff_means.boot$t)) +
  xlab("mean(ilec) - mean(clec)")

qqnorm(diff_means.boot$t)
qqline(diff_means.boot$t)
```





mean$(\hat{\theta}^* - \hat{\theta})$

```
# Your turn: estimate the bias and se of the sampling distribution
```

sd$(\hat{\theta}^*)$

Which confidence intervals should we use? BCa

```
# Your turn: get the chosen CI using boot.ci
```

Is there evidence that

$$H_0 : \mu_1 - \mu_2 = 0$$
$$H_a : \mu_1 - \mu_2 < 0$$

is rejected?

Yes!

2 Parametric Bootstrap

In a **nonparametric bootstrap**, we resample the data

Create a bootstrap sample $y_1^*, \ldots, y_n^*$ iid from the empirical dsn $\hat{F}$.

↳ this is equivalent to resampling the original data w/ replacement.

In a **parametric bootstrap**, we assume a parametric model.

Key idea: use a fitted parametric model $\hat{F}(y) = F(y \mid \hat{\psi})$ to estimate F where $\hat{\psi}$ is estimated using MLE (or some other method) from data.

create a bootstrap sample $y_1^*, \ldots, y_n^*$ iid from $F(y \mid \hat{\psi})$, i.e. resample from a model w/ parameters are estimated using original data.

For both methods,

1. We compute the statistic $\hat{\theta}^{*(b)}$ for each bootstrap sample $y_1^{*(b)}, \ldots, y_n^{*(b)}$

2. We repeat the procedure B times to get

$$\hat{\theta}^{*(1)}, \ldots, \hat{\theta}^{*(B)}$$

and make inferences using the result.

2.1 Bootstrapping for linear regression

Consider the regression model $Y_i = \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i, i = 1, \ldots, n$ with $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$.

$Y_1, \ldots, Y_n$ not iid! They have different conditional means

Resampling in the bootstrap must be done on iid quantities!

Two approaches for bootstrapping linear regression models –

1. Bootstrapping the residuals (model based resampling) – parametric.
2. Paired bootstrapping (case resampling) – nonparametric.

2.1.1 Bootstrapping the residuals

1. Fit the regression model using the original data
2. Compute the residuals from the regression model,
$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}}, \quad i = 1, \ldots, n$$
3. Sample $\hat{\epsilon}_1^*, \ldots, \hat{\epsilon}_n^*$ with replacement from $\hat{\epsilon}_1, \ldots, \hat{\epsilon}_n$.
4. Create the bootstrap sample
$$y_i^* = \boldsymbol{x}_i^T \hat{\boldsymbol{\beta}} + \epsilon_i^*, \quad i = 1, \ldots, n$$
5. Estimate $\hat{\boldsymbol{\beta}}^*$
6. Repeat steps 2-4 B times to create B bootstrap estimates of $\hat{\beta}$.

Assumptions:

2.1.2 Paired bootstrapping

Resample $z_i^* = (y_i, \boldsymbol{x}_i)^*$ from the empirical distribution of the pairs $(y_i, \boldsymbol{x}_i)$.

Assumptions:

2.1.3 Which to use?

1. Standard inferences -

2. Bootstrapping the residuals -

3. Paired bootstrapping -

Your Turn

This data set is the Puromycin data in R. The goal is to create a regression model about the rate of an enzymatic reaction as a function of the substrate concentration.

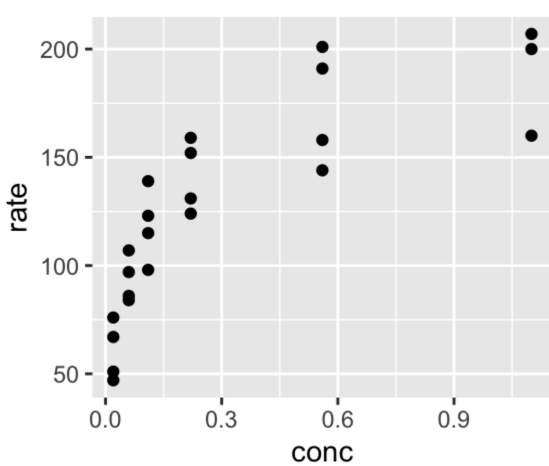
```
head(Puromycin)
```

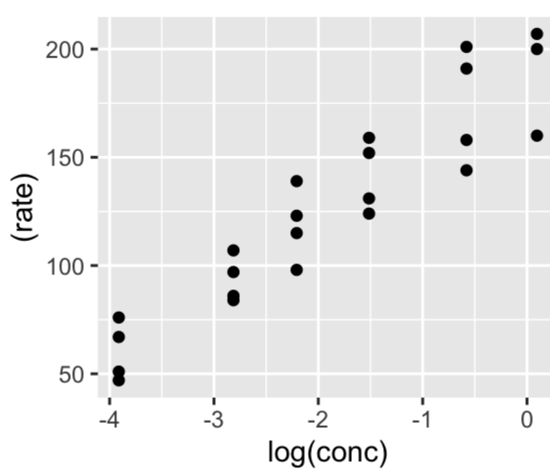
```
##   conc rate   state
## 1 0.02   76 treated
## 2 0.02   47 treated
## 3 0.06   97 treated
## 4 0.06  107 treated
## 5 0.11  123 treated
## 6 0.11  139 treated
```

```
dim(Puromycin)
```

```
## [1] 23  3
```

```
ggplot(Puromycin) +
  geom_point(aes(conc, rate))

ggplot(Puromycin) +
  geom_point(aes(log(conc), (rate)))
```





2.1.4 Standard regression

```
m0 <- lm(rate ~ conc, data = Puromycin)
plot(m0)
summary(m0)
```

```
##
## Call:
## lm(formula = rate ~ conc, data = Puromycin)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.861 -15.247  -2.861  15.686  48.054
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    93.92       8.00   11.74 1.09e-10 ***
## conc          105.40      16.92    6.23 3.53e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.82 on 21 degrees of freedom
## Multiple R-squared:  0.6489, Adjusted R-squared:  0.6322
## F-statistic: 38.81 on 1 and 21 DF,  p-value: 3.526e-06
```

```
confint(m0)
```

```
##                2.5 %   97.5 %
## (Intercept) 77.28643 110.5607
## conc        70.21281 140.5832
```

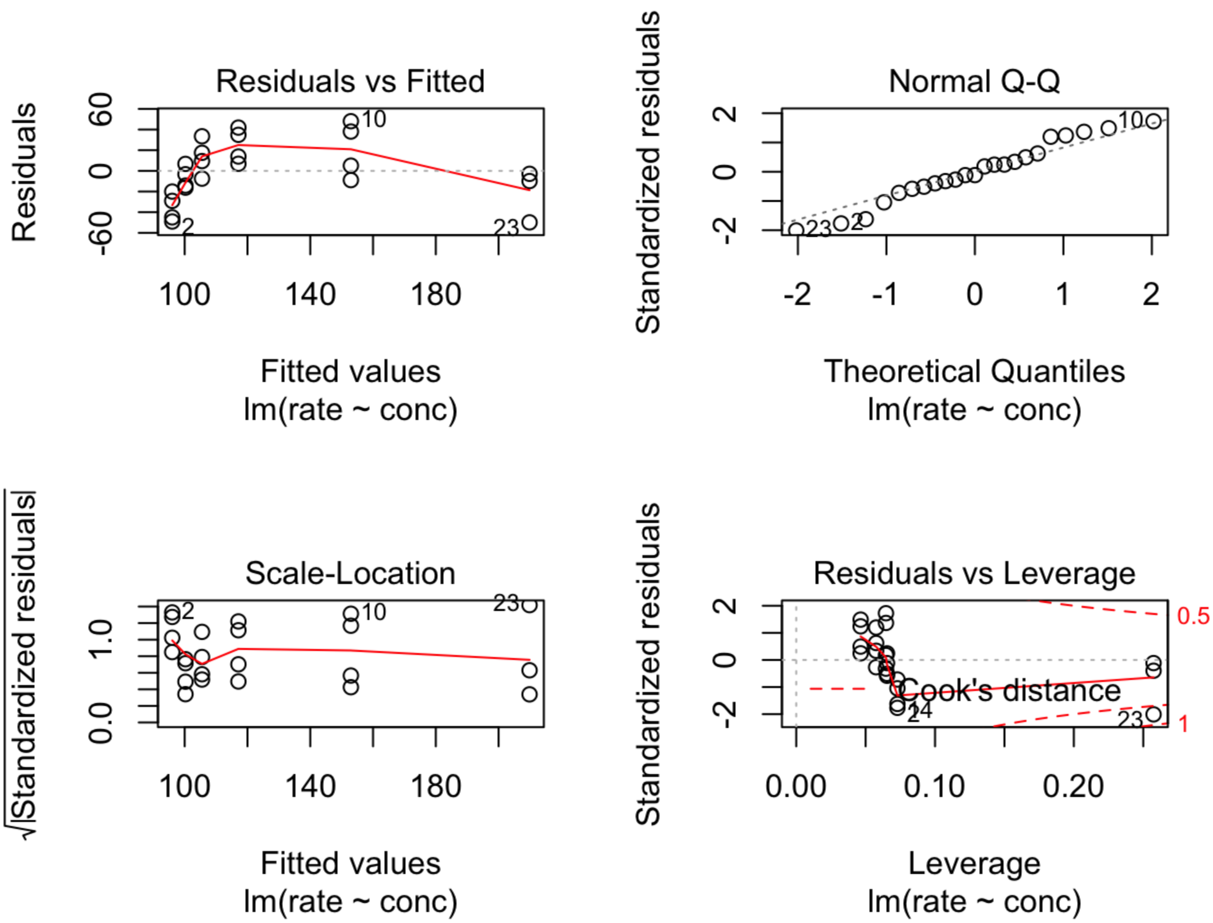
```
m1 <- lm(rate ~ log(conc), data = Puromycin)
plot(m1)
summary(m1)
```

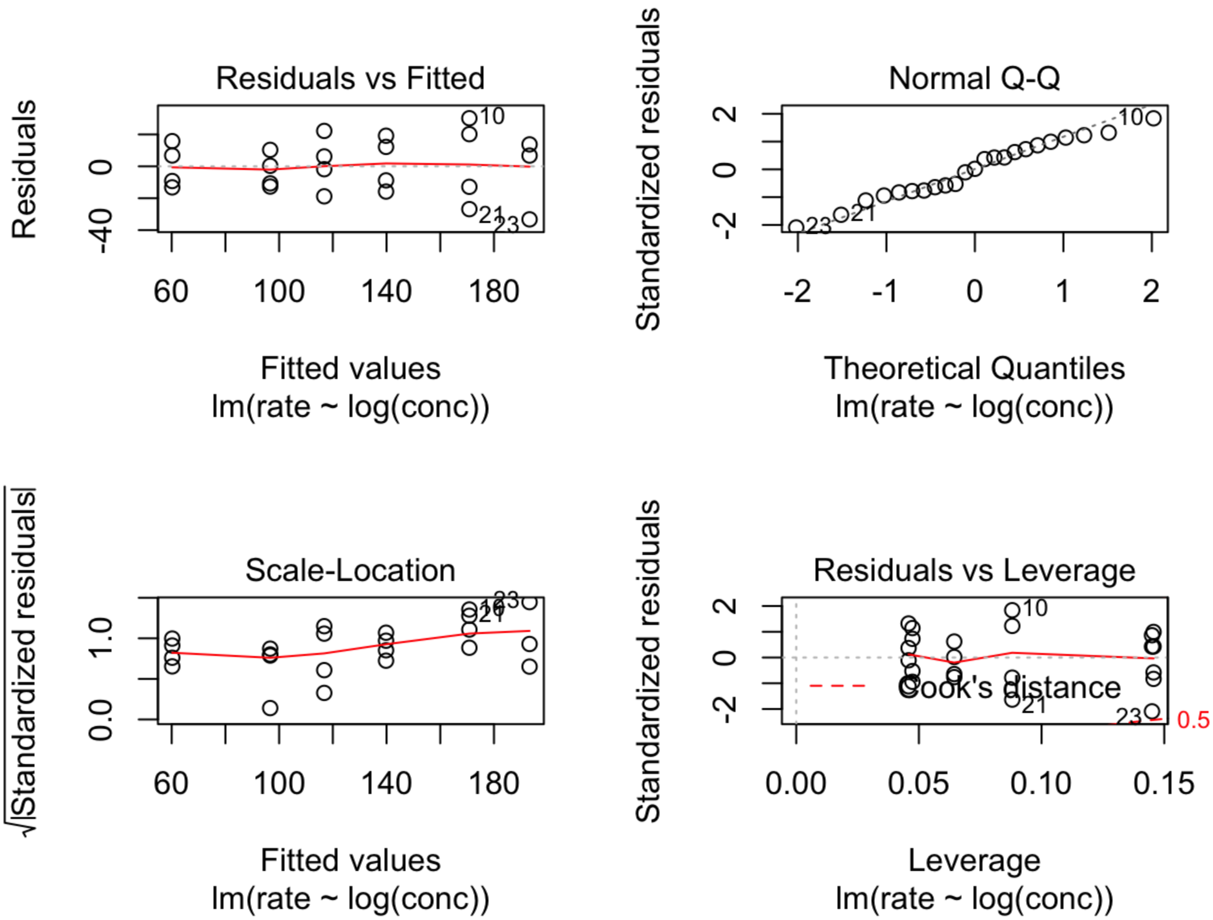
```
##
## Call:
## lm(formula = rate ~ log(conc), data = Puromycin)
##
```

```
## Residuals:
##      Min       1Q  Median       3Q      Max
## -33.250 -12.753   0.327  12.969  30.166
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  190.085      6.332   30.02  < 2e-16 ***
## log(conc)     33.203      2.739   12.12 6.04e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17.2 on 21 degrees of freedom
## Multiple R-squared:  0.875,  Adjusted R-squared:  0.869
## F-statistic: 146.9 on 1 and 21 DF,  p-value: 6.039e-11
```

```
confint(m1)
```

```
##                 2.5 %   97.5 %
## (Intercept) 176.91810 203.2527
## log(conc)    27.50665  38.8987
```


Residuals vs Fitted
Residuals
Fitted values
lm(rate ~ conc)
Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(rate ~ conc)
Scale-Location
√|Standardized residuals|
Fitted values
lm(rate ~ conc)
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage
lm(rate ~ conc)



2.1.5 Paired bootstrap

```
# Your turn
library(boot)

reg_func <- function(dat, idx) {
  # write a regression function that returns fitted beta
}

# use the boot function to get the bootstrap samples

# examing the bootstrap sampling distribution, make histograms

# get confidence intervals for beta_0 and beta_1 using boot.ci
```

2.1.6 Bootstrapping the residuals

```r
# Your turn
library(boot)

reg_func_2 <- function(dat, idx) {
  # write a regression function that returns fitted beta
  # from fitting a y that is created from the residuals

}

# use the boot function to get the bootstrap samples

# examing the bootstrap sampling distribution, make histograms

# get confidence intervals for beta_0 and beta_1 using boot.ci
```