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

+ which 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).

estimate sampling dsn of $\hat{\theta}$ via bootstrap.

$\hat{\theta}^*$

$\alpha/2$ $1-\alpha$ $\alpha/2$

? ?

1.4.1 Percentile Bootstrap CI
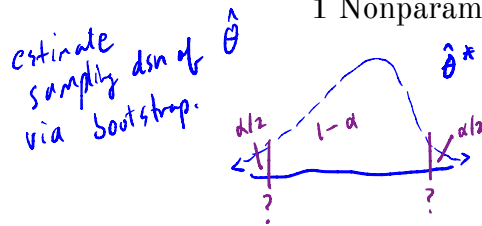
Let $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}$ be bootstrap replicates and let $\hat{\theta}_{\alpha/2}$ be the $\alpha/2$ quantile of $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}$.

estimate of sampling dsn "bootstrap dsn"

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(`$\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(B)}$`)`, the percentile CI is

vector of bootstrap replicates.

```
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 dsn.

③ Drawback: CI's usually too narrow! (coverage too 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

simplified.

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

recentering the interval based on estimated bias.

$1-\alpha/2$ quantile of bootstrap dsn for $\hat{\theta}$

estimate of θ from original sample.

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

Assumptions/usage

(1) Better than percentile bootstrap b/c corrects for bias.
(does nothing for skewness).

(2) Harder to explain.

1.4.3 Standard Normal Bootstrap CI

From the CLT,

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

under some assumptions...

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

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

comes from bootstrap replicates.

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

Assumptions/usage

① $\hat{\theta} \ \tilde{\sim} \ N\left(E(\hat{\theta}), \, se(\hat{\theta})^2\right) \longleftarrow$ BIG assumption if $\hat{\theta}$ is not a sample mean.

② If $\hat{\theta}$ is unbiased $\Rightarrow E(\hat{\theta}) = \theta \ \checkmark$
(If not unbiased, bias corrected version w/ this method too) $\rightarrow$ see later code.

③ typically requires large n.

this is a 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 $se(\hat\theta)$).*

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

is not $s/\sqrt{n}$!

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

So we cannot claim $t^* \sim t_{n-1}$

$\Rightarrow$ 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.

studentized. *bootstrap*

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

quantile of the bootstrap "t-type" statistic.

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

estimate of θ from original sample   *?*

$\hat{se}(\hat\theta) = $ *se based on* $(\hat\theta^{(1)}, .., \hat\theta^{(B)})$.

read "se of this" s standin for θ

Overview

t-type statistic

$$t^{(1)} = \frac{\hat\theta^{(1)} - \hat\theta}{\hat{se}(\hat\theta^{(1)})} \;,\; \cdots \;,\; t^{(B)} = \frac{\hat\theta^{(B)} - \hat\theta}{\hat{se}(\hat\theta^{(B)})}.$$

= bootstrap estimate of se of $\hat\theta$ based on the first bootstrap sample

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

1. Compute $\hat\theta$ [based on sample $X_1, .., X_n$].

2. For each replicate $b = 1, .., B$
   a) Sample w/ replacement from $\underline{X}$
   $$\underline{X}^{(b)} = (X_1^{(b)}, .., X_n^{(b)}).$$
   b) Compute $\hat\theta^{(b)}$.
   c) For each replicate $r = 1, .., R$
      i) sample w/ replacement from $\underline{X}^{(b)}$
      $$\underline{X}^{(b)(r)} = (X_1^{(b)(r)}, .., X_n^{(b)(r)})$$
      ii) Compute $\hat\theta^{(b)(r)}$.
   d) Compute $\hat{se}(\hat\theta^{(b)}) = sd(\hat\theta^{(b)(1)}, .., \hat\theta^{(b)(R)})$

DOUBLE BOOTSTRAP!
WOAH!

   e) Compute "t style" statistics
   $$t^{(b)} = \frac{\hat\theta^{(b)} - \hat\theta}{\hat{se}(\hat\theta^{(b)})}.$$

3. get quantiles $t^*_{1-\alpha/2}, t^*_{\alpha/2}$

4. Compute CI.

Assumptions/usage

① Require small bias and skewness in bootstrap dsn.

✳ ② Computationally intensive.

③ Need $\hat{\theta}$ independent 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.

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

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

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 percentile bootstrap, but instead of

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

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

Assumptions / usage

① Better theoretical & practical performance than percentile method (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 "Incum-
bent" local exchange carrier.

[handwritten: ↑ other carriers   ↑ verizon customers]

[handwritten: Verizon required by law to serve both at same speed.]

```r
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
```

```r
Verizon %>%
  group_by(Group) %>%
  summarize(mean = mean(Time), sd = sd(Time), min = min(Time), max =
  max(Time)) %>%
  kable()
```
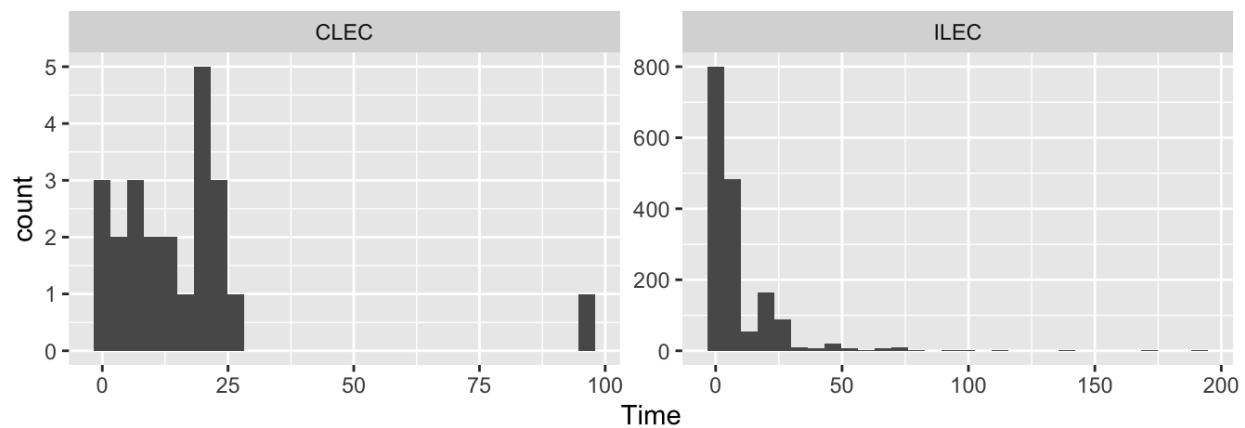
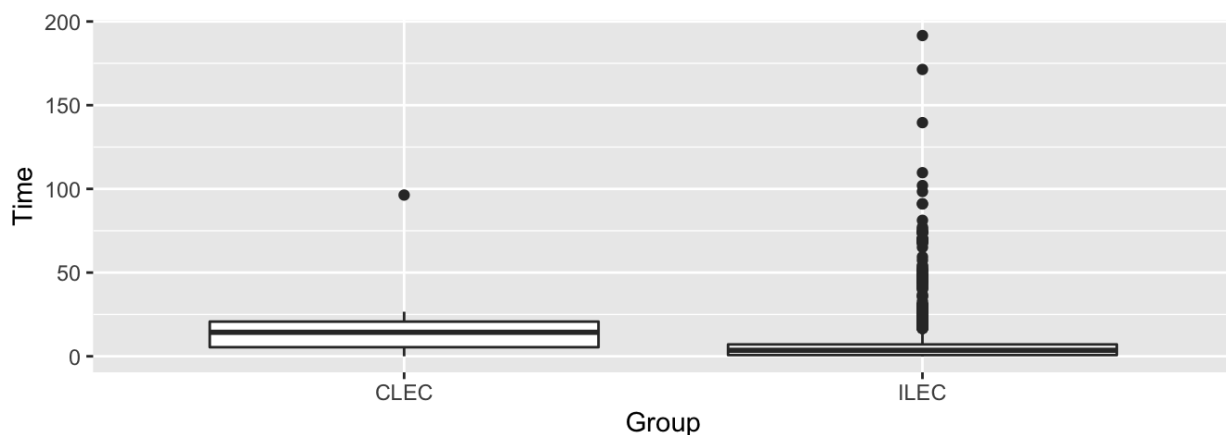| Group	mean	sd	min	max	[handwritten: n]
CLEC	16.509130	19.50358	0	96.32	[handwritten: 23]
ILEC	8.411611	14.69004	0	191.60	[handwritten: 1664]

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

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


1.5 Bootstrapping CIs

also:
simpleboot

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

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

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

plot(boot.ilec)
```
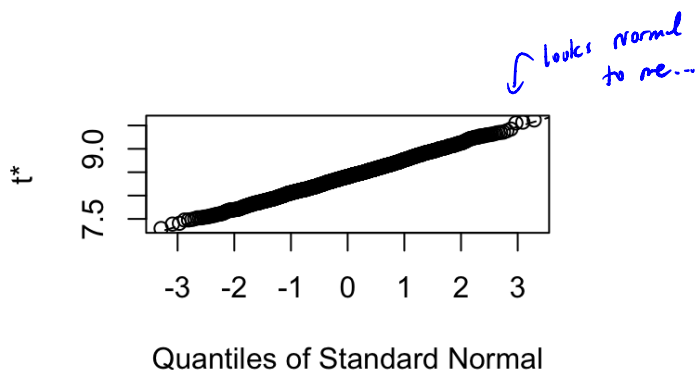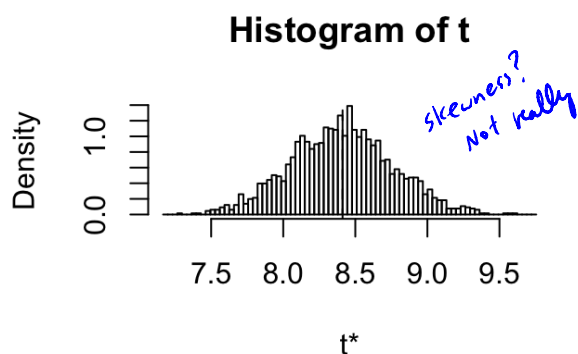
(handwritten annotations:) function. → ; data; index of resampled observations; $\hat{\theta}^*$; just Verizon customers.; data; statistic function.; R # of replicates

Histogram of t

(handwritten annotations on plots:) skewness? Not really.; looks normal to me...

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

(handwritten annotation:) missing studentized, later.

```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 (handwritten)

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

bootstrap samples of $\hat{\theta}$: $\hat{\theta}^{(1)}, \ldots, \hat{\theta}^{(R)}$ (handwritten)

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

$\hat{\theta}$ (handwritten) *$\bar{\theta}^*$* (handwritten) *bias corrected version* (handwritten)

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

bias corrected (handwritten)

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

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

very similar ⇒ not much bias. (handwritten)

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

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

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

Double bootstrap!

[bracket pointing down]

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

```r
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")
```

[annotations:] estimate w/ bootstrap.

$$Var(\bar{x}) = \frac{Var\, X}{n}$$

This not always so easy.

also look at simpleboot: one.boot

```
## 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? *dsn*

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

> *BCa my default choice because has been shown to have good coverage.*

> *Percentile & Basic not a bad choice if explaining to stake holders.*

n large + QQ plot ⟹ Normal is a valid interval as well.

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 $b = 1, ..., B$

    a) Resample w/ replacement a sample of size n from sample 1 and a sample of size m from sample 2.

    b) Compute a statistic that compares two groups (i.e. $\hat{\theta}^{(b)} = \bar{x}_1^{(b)} - \bar{x}_2^{(b)}$)

2. Construct a bootstrap dsn of the statistic $\hat{\theta}^{(1)}, ..., \hat{\theta}^{(B)}$ — inspect shape, $\hat{bias}$, $\hat{se}$

3. Compute an 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)
```
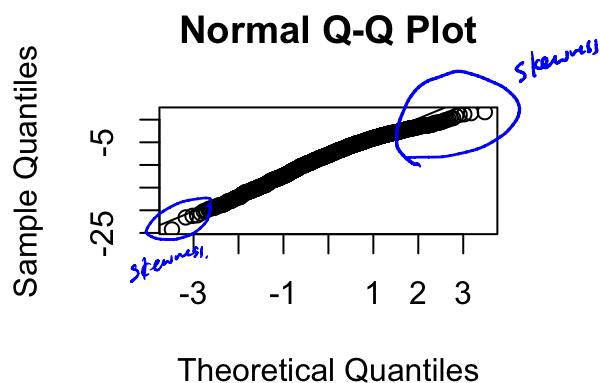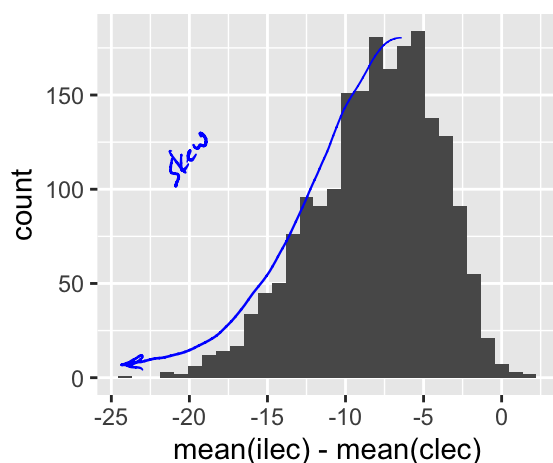
non-verizon customer response times.

verizon *non-verizon*

plot bootstrap dsn

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

Which confidence intervals should we use?

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

Is there evidence that

$$ILEC \sim CLEC$$
$$H_0 : \mu_1 - \mu_2 = 0$$
$$H_a : \mu_1 - \mu_2 < 0$$

is rejected?

Yes!

There is evidence for the alternative, Verizon is treating their customers differently than other company's customers!

2 Parametric Bootstrap

In a **nonparametric bootstrap**, we resample observed data.

Create a bootstrap sample $y_1^*, \ldots, y_n^*$ iid from empirical distribution $\hat{F}$. 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 $\hat{F}(y) = F(y|\hat{\gamma})$ to estimate F where $\hat{\gamma}$ is estimated from the data (using MLE).

Create a bootstrap sample $y_1^*, \ldots, y_n^*$ iid from $F(y|\hat{\gamma})$, i.e. resample from a model w/ parameters estimated by original sample 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

can get more complicated

$$Y_i \sim N(x_i^T \beta, \sigma^2)$$

independently.

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

$Y_1, \ldots, Y_n$ *independent but* <u>*not*</u> *iid! They have different conditional means!*

Resampling in the bootstrap must be done on <u>*iid*</u> *quantities!*

Two approaches for bootstrapping linear regression models –

1. *Bootstrap the residuals (model based bootstrapping) – parametric.*

2. *Paired bootstrapping (case resampling) – non parametric.*

2.1.1 Bootstrapping the residuals

1. Fit the regression model using the original data

2. Compute the residuals from the regression model,

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - x_i^T \hat{\beta}, \quad i = 1, \ldots, n \qquad \text{\textit{residuals } } \epsilon_i \text{ \textit{are assumed iid} } \checkmark$$

3. Sample $\hat{\epsilon}_1^*, \ldots, \hat{\epsilon}_n^*$ with replacement from $\hat{\epsilon}_1, \ldots, \hat{\epsilon}_n$.

4. Create the bootstrap sample

fitted values based on model/data

$$y_i^* = \underbrace{x_i^T \hat{\beta}}_{} + \hat{\epsilon}_i^*, \quad i = 1, \ldots, n$$

resampled residuals.

5. Estimate $\hat{\beta}^*$ $\{y_i^*, x_i\}$

use $\{y_1^*, x_1\} \\ \{y_n^*, x_n\}$ to fit new regression model, get $\hat{\beta}^*$

6. Repeat steps 2-4 B times to create B bootstrap estimates of $\hat{\beta}$.

to estimate dsn of $\hat{\beta}$.

Assumptions:

The design matrix $X = (x_1, \ldots, x_n)$ is fixed.

$*$ ϵ_i are iid.

2.1.2 Paired bootstrapping (Case resampling).

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

Fit regression model w/ n bootstrapped pairs $(y_i, \underline{x}_i)^*$

$$y_i^* = (\underline{x}_i^*)^T \beta + \varepsilon_i \qquad i = 1, \cdots, n$$

Assumptions:

Assumes (y_i, x_i) are iid from population.

Can have varying design matrix X.

2.1.3 Which to use?

1. Standard inferences - (i.e. STAT 341)

   Most of the time.

2. Bootstrapping the residuals -
   - most appropriate for designed experiments where $\underline{X}_i$ fixed in advance.
   - model-based (regression model must be valid for the data).
   - often useful if complex sampling distribution for $\hat{\beta}$.

3. Paired bootstrapping -
   - robust to model mis-specification
     (if you have doubts about the adequacy of the regression model such as heteroscedasticity).
   - useful for observational studies where values of explanatory variables aren't fixed in advance $\implies$ paired bootstrap mirrors data generating mechanism.

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.

```r
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
```
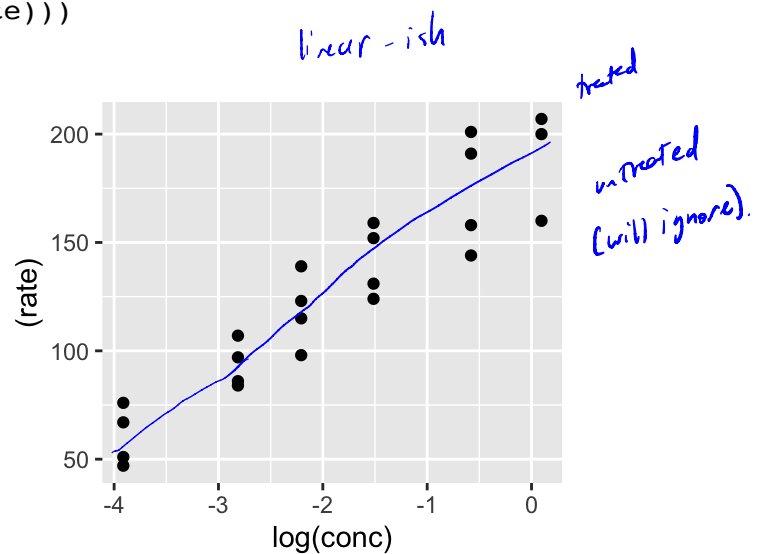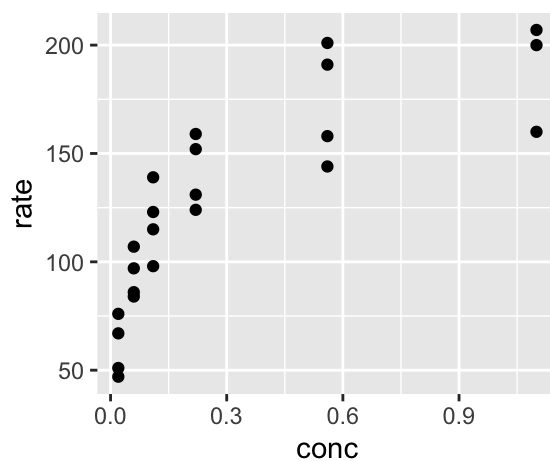
```r
dim(Puromycin)
```

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

n=23, small data.

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

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

linear-ish

treated

untreated (will ignore).

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)
##
```
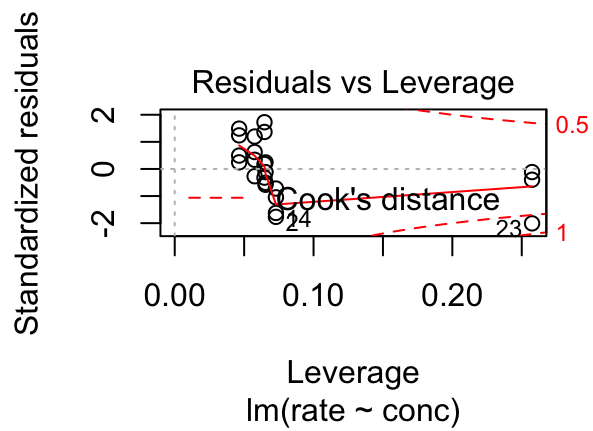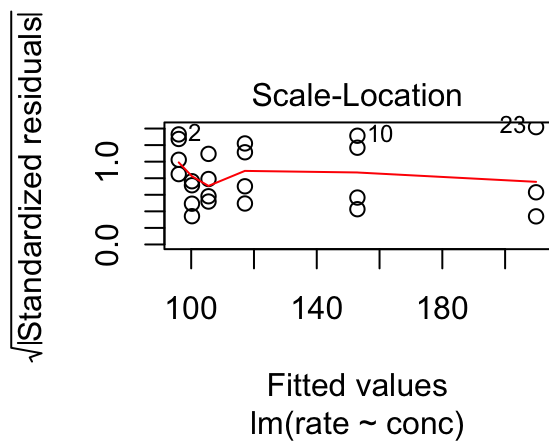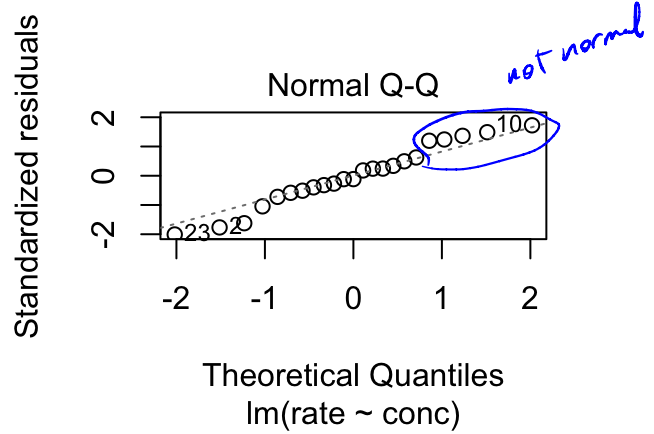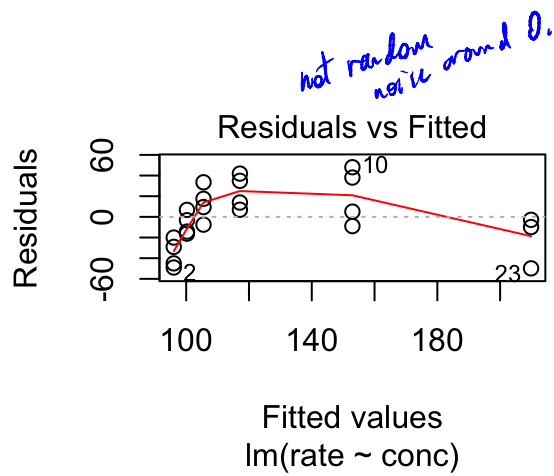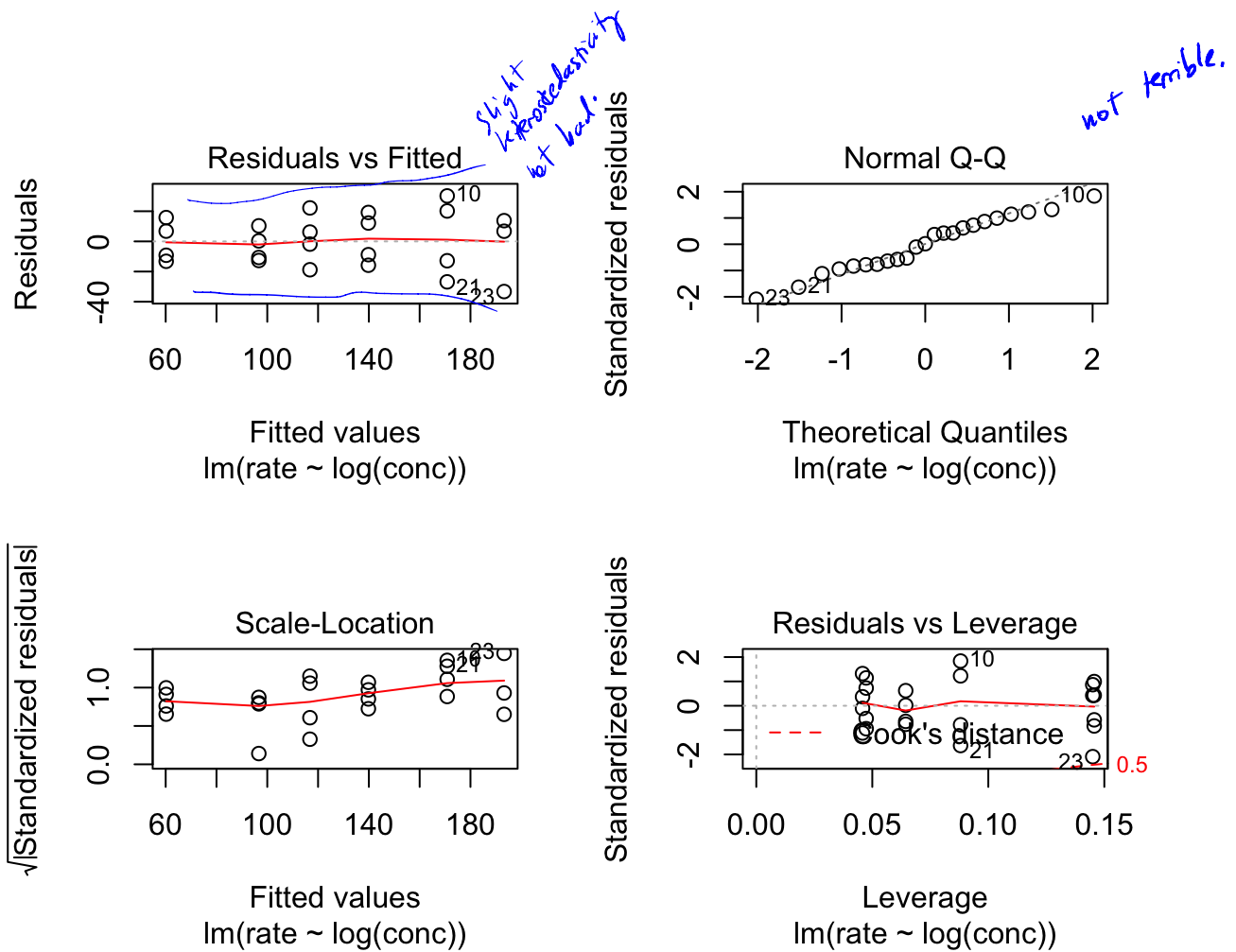
```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.250 -12.753   0.327  12.969  30.166
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  190.085      6.332   30.02  < 2e-16 ***
## log(conc)     33.203      2.739   12.12 6.04e-11 ***
## ---
## Signif. 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 — lm(rate ~ log(conc))

[handwritten: slight heteroscedasticity not bad.]

Normal Q-Q — lm(rate ~ log(conc))

[handwritten: not terrible.]

Scale-Location — lm(rate ~ log(conc))

Residuals vs Leverage — lm(rate ~ log(conc))

2.1.5 Paired bootstrap

```r
# 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
```

results are very similar to standard inference. => OK to use standard results.

If bootstrapping,

this is designed experiment

so bootstrapping residuals is good choice as long as model fit looks appropriate (which it did).