

The Bootstrap

0.1 The plug-in principle for finding estimators

Under a parametric model $\mathcal{P} = \{P_\theta; \theta \in \Theta\}$ (or a non-parametric $\mathcal{P} = \{P_F; F \in \mathcal{F}\}$), any real-valued characteristic τ of a particular member P_θ (or P_F) can be written as a mapping from the parameter-space Θ , i.e. $\tau : \Theta \mapsto \mathbf{R}$. If your observations y comes from P_{θ_0} and you have derived an estimate $\hat{\theta} \in \Theta$ of θ_0 (for example by Maximum-Likelihood), it is natural to use $\tau(\hat{\theta})$ as an estimate of $\tau(\theta_0)$. This method for constructing estimates is commonly referred to as *the plug-in principle*, since we “plug” the estimate $\hat{\theta}$ into the mapping $\tau(\cdot)$.

Example 0.1 (Independent normals cont.). Assume you have observations as in Example ??, and you are interested in the probability that a new observation $Y_{n+1} \in \mathbf{R}$ from the same distribution exceeds a level u , i.e. $\tau(\theta_0) = \Phi((u - \mu_0)/\sigma_0)$. If you have derived the Maximum-Likelihood estimate $\hat{\theta} = (\hat{\mu}, \hat{\sigma}^2)$ of θ_0 , $\tau(\theta_0)$ should be estimated by $\tau(\hat{\theta}) = \Phi((u - \hat{\mu})/\hat{\sigma})$, following the plug-in principle. Similarly, if you want to estimate the error distribution of $\hat{\mu} = \bar{y}$ then $\tau(\theta_0) = F_\Delta(u) = P(\bar{Y} - \mu_0 \leq u) = \Phi(u/(\sigma_0/\sqrt{n}))$ and the plug-in estimate $\tau(\hat{\theta}) = \Phi(u/(\hat{\sigma}/\sqrt{n}))$.

Maximum-Likelihood estimators generally work well for parametric models, for non-parametric models the natural choice of an estimator for F is the empirical distribution function:

Definition 0.1. The empirical distribution derived from a sample $y = (y_1, \dots, y_n)$, is the uniform distribution on the set $\{y_1, \dots, y_n\}$ with distribution function

$$\hat{F}(u) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{y_i \leq u\}. \quad (1)$$

As before we interpret the inequalities pointwise if y_i are vectors. Importantly, a random variable $Z \in \mathbf{R}$ distributed according to \hat{F} is discrete and satisfies $P(Z = y_i) = 1/n$, $i = 1, \dots, n$ if all the values in $\{y_1, \dots, y_n\}$ are distinct.

Example 0.2. The plug-in principle also applies with the empirical distribution function as argument. Here we assume $y = (y_1, \dots, y_n) \in \mathbf{R}^n$ is an observation of $Y = (Y_1, \dots, Y_n)$, a vector of independent random variables with common distribution function F_0 and that \hat{F} is the empirical distribution function.

- *The expected value:* If we want to estimate

$$\tau(F_0) = E(Y_1) = \int u dF_0(u),$$

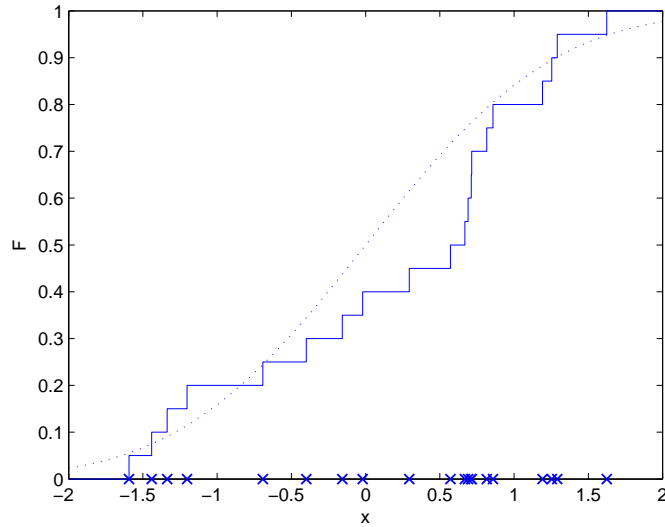


Figure 1: Empirical distribution function based on 20 draws from $N(0, 1)$, the dotted line is true distribution, draws represented by \times :es on the x -axis.

the plug-in estimator is

$$\tau(\hat{F}) = \int u d\hat{F}(u) = n^{-1} \sum_{i=1}^n y_i,$$

i.e. the arithmetic mean.

- *The variance:* If we want

$$\tau(F_0) = V(Y_1) = E(E(Y_1) - Y_1)^2 = \int \left(\int u dF_0(u) - v \right)^2 dF(v),$$

the plug-in estimator is

$$\tau(\hat{F}) = \int \left(\int u d\hat{F}(u) - v \right)^2 d\hat{F}(v) = n^{-1} \sum_{i=1}^n (\bar{y} - y_i)^2.$$

- *Quantiles:* If we want

$$\tau(F_0) = F_0^{-1}(p) = \inf\{u; F_0(u) \geq p\},$$

the plug-in estimator is

$$\tau(\hat{F}) = \inf\{u; \hat{F}(u) \geq p\} = y_{(\lceil np \rceil)},$$

the $\lceil np \rceil$:th largest value in $\{y_1, \dots, y_n\}$.

0.2 The plug-in principle for evaluating estimators

Now we have constructed an estimator $t(\cdot)$ for τ using the plug-in principle (or some other principle), and want to assess its uncertainty. As we have seen, estimating the error-distribution F_Δ is often crucial here. But, under model \mathcal{P} , F_Δ is of course uniquely determined by θ_0 (or F_0) and it is natural to estimate also this function by its plug-in estimate. I.e. since F_Δ is the distribution function of $\Delta(Y) = t(Y) - \tau(P_0)$, $Y \sim P_0$, we can write its plug-in estimate as F_{Δ^*} , the distribution function of $\Delta(Y^*) = t(Y^*) - \tau(\hat{P})$, $Y^* \sim \hat{P}$. The plug-in principle has its limitations here though. It is only for very special models (like the Normal) and simple estimators t that F_{Δ^*} can be computed explicitly. We have already seen in (??) how computing $F_\Delta(u)$ required an n -dimensional integral over a complicated set. This is where Monte-Carlo integration comes into the picture; even if we can't compute F_{Δ^*} it is often easy to simulate from \hat{P} (and hence from F_{Δ^*}).

Plug-in + Monte-Carlo: The Bootstrap

Algorithm 0.1 (The Bootstrap algorithm).

1. *Estimation:* Use data y to construct an estimate \hat{P} of P_0 .
2. *Simulation:* Draw B independent samples $y^{*b} \in \mathcal{Y}$, $b = 1, \dots, B$ from the distribution \hat{P} .
3. *Approximation:* Compute $t^{*b} = t(y^{*b})$, $b = 1, \dots, B$, and use these values to approximate e.g. one of the following plug-in estimates:

- (a) The error distribution function

$$P(t(Y^*) - \tau(\hat{P}) \leq u) = F_{\Delta^*}(u) \approx \frac{1}{B} \sum_{b=1}^B \mathbf{1}\{t^{*b} - \tau(\hat{P}) \leq u\}.$$

- (b) Quantiles of the error distribution

$$F_{\Delta^*}^{-1}(p) \approx \Delta^{*([\![pB]\!]])} = t^{*([\![pB]\!]])} - \tau(\hat{P}),$$

the $[\![pB]\!]$:th largest value of t^{*b} , $b = 1, \dots, B$.

- (c) The variance of $t(Y^*)$,

$$\text{Var}(t(Y^*)) \approx \frac{1}{B} \sum_{b=1}^B (t^{*b} - \bar{t}^*)^2.$$

- (d) The bias of t

$$E(t(Y^*)) - \tau(\hat{P}) \approx \bar{t}^* - \tau(\hat{P})$$

- (e) ...

Most often, in the above algorithm, $\tau(\hat{P}) = t(y)$. If we are unable to derive $\tau(\hat{P})$ explicitly it can be approximated with a similar procedure.

In the following sections we will discuss how to perform Steps 1-2 under some common modelling assumptions. First a basic example.

Example 0.3 (Airconditioning). The file `aircon.mat` contains 12 times between failures for airconditioning equipment in a Boeing aircraft. A reasonable assumption is that

data are independent with an $\text{Exp}(\theta)$ distribution. We estimate θ with the arithmetic mean $t(y) = \bar{y} = 106.4$. In the left panel of Figure ?? we have plotted the $\text{Exp}(106.4)$ distribution function together with the empirical distribution function, while the fit is not perfect the deviation from the Exponential distribution does not look alarming. We want to construct an upper 95% confidence interval for θ based on t , looking at (??), we want to set $L(y) = 0$, $U(y) = 106.4 - F_{\Delta}^{-1}(0.05)$ and hence need an estimate of $\tau = F_{\Delta}^{-1}(0.05)$. Using the bootstrap we proceed as follows in Matlab

```
for b=1:10000
    tstar(b)=mean(exprnd(106.4,1,12));
end
```

which draws a sample of 10000 from T^* , the arithmetic mean of twelve $\text{Exp}(106.4)$ variates. A histogram is shown in the right panel of Figure ?. The plug-in estimate of θ is obviously $t = 106.4$, hence

```
Delta=tstar-106.4;
```

is a bootstrap draw from the error distribution and τ can be estimated by the empirical 5% quantile

```
quantile(Delta,.05)
ans =
    -44.8750
```

and finally, our confidence interval is given by $[0, 151.3]$. An interval based on Normal approximation is

```
106.4-106.4*norminv(.05)/12^(1/2)
ans =
    156.9217
```

where we used that the Exponential distribution has both mean and standard deviation θ .

If we are interested in the failure intensity $1/\theta$, this can be estimated by $1/t(y) = 0.0094$. However, this is a biased estimate. The bias is given by $E(1/t(Y)) - 1/\theta$ and can be estimated by the plug-in estimate $E(1/t(Y^*)) - 1/t(y)$, in Matlab

```
bias=mean(1./tstar)-1/106.4
bias =
    8.9401e-004
```

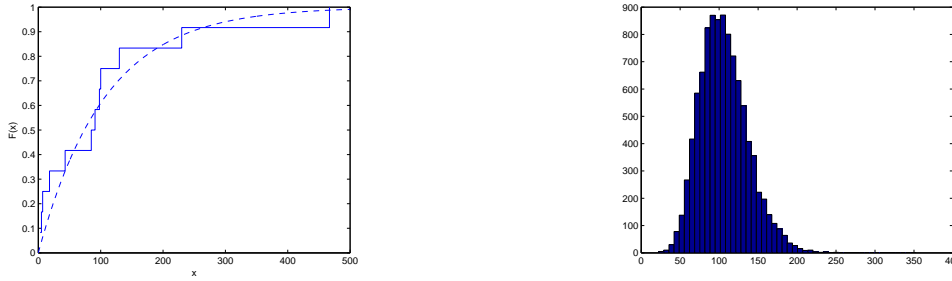


Figure 2: Left panel: Fitted Exponential cdf (dashed) and empirical distribution (solid). Right panel: Histogram of $t(y^*)$.

and we might want to report the bias-corrected estimate $0.0094 - 0.0009 = 0.0085$ rather than our original choice.

0.3 The non-parametric Bootstrap for i.i.d. observations

Here we look at the model corresponding to Example ??, though with possibly vector-valued y_i . That is, $y = (y_1, \dots, y_n)$ is an observation of $Y = (Y_1, \dots, Y_n)$, where the Y_i 's, $Y_i \in \mathbf{R}^m$, are independent random vectors with common distribution function $F_0 : \mathbf{R}^m \mapsto [0, 1]$. We make no assumptions on the shape of F_0 .

We have already introduced the empirical distribution function \hat{F} as the natural estimate of F . Since the distribution function of Y is $\prod_{i=1}^n F_0(u_i)$ we let \hat{P} be the probability distribution with distribution function $\prod_{i=1}^n \hat{F}(u_i)$. This was Step 1. In Step 2 we need to draw samples $y^{*b} = (y_1^{*b}, \dots, y_n^{*b})$ from this distribution, but this is easy: Firstly, the y_i^{*b} 's should be drawn independently of each other. Secondly, the empirical distribution function is the uniform distribution on $\{y_1, \dots, y_n\}$, hence we just draw n values from this set randomly with replacement.

Algorithm 0.2 (Drawing y^* from $\prod_{i=1}^n \hat{F}(u_i)$).

1. Let \hat{F} be the empirical distribution function of a sample $y = (y_1, \dots, y_n)$.
2. Draw i_1, i_2, \dots, i_n independently from the uniform distribution on the set of integers $\{1, 2, \dots, n\}$.
3. $y^* = (y_{i_1}, y_{i_2}, \dots, y_{i_n})$ is now a draw from $\prod_{i=1}^n \hat{F}(u_i)$.

Example 0.4 (Airconditioning cont.). Lets return to the data in Example ??, but this time we will be reluctant to assume data are from the Exponential distribution. We will still assume the failure-times are independent and from the same distribution though, and we are still interested in the expected failure time. Hence, failure times are independent with unknown distribution F_0 . We estimate F_0 by the empirical distribution function, plotted in Figure ??, and proceed as in Example ??, with the difference that new samples are drawn from $\prod_{i=1}^{12} \hat{F}(u_i)$ (here `ac` is the Matlab vector containing data):

```
for b=1:10000
    i=ceil(12*rand(1,12));
    tstar(b)=mean(ac(i));
end
```

a histogram of the sampled means is given in Figure ??, it looks slightly wider than the corresponding plot in Figure ?? and indeed

```
Delta=tstar-106.4;
quantile(Delta,.05)
```

ans =

```
-52.7750
```

gives a wider confidence interval $[0, 159.2]$. This is not surprising since it is more difficult to estimate the expected value under the larger non-parametric model.

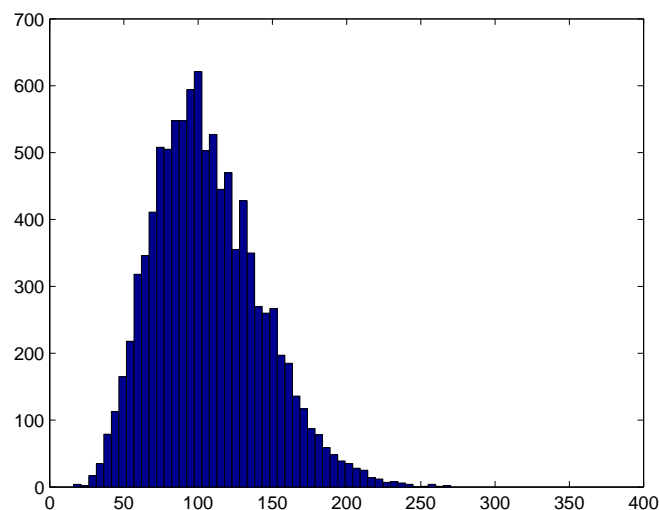


Figure 3: Histogram of bootstrap correlations from Example ??

Example 0.5. The above method applies almost as easily to slightly more complicated data sets. We consider here a set of data relating two score tests, LSAT and GPA, at a sample of 15 American law schools. Of interest is the correlation between these measurements. The data are given as

LSAT	GPA
576	3.39
635	3.30
558	2.81
578	3.03
666	3.44
580	3.07
555	3.00
661	3.43
651	3.36
605	3.13
653	3.12
575	2.74
545	2.76
572	2.88
594	2.96

and are plotted in Figure ???. They are stored in matrix form in the file `matrix law.mat`. Estimating the population correlation by the sample correlation gives $\hat{\theta} = 0.776$, but how

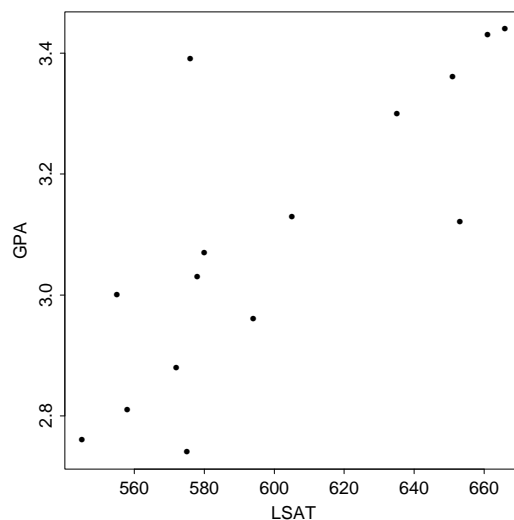


Figure 4: Scatterplot of law school data

accurate is this estimate? Our model in this case is that the score-pairs are independent

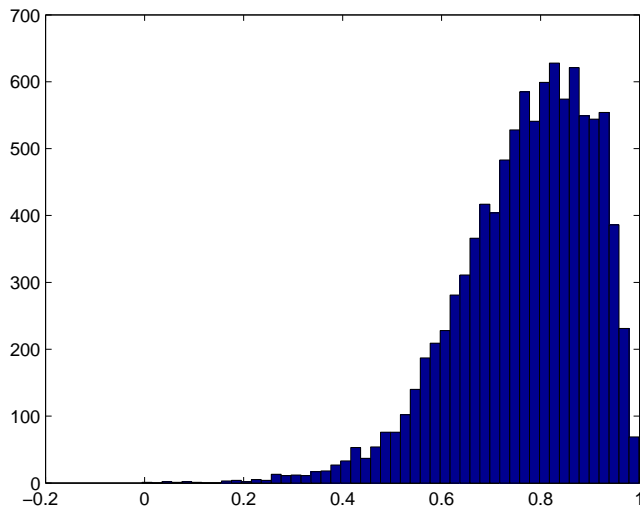


Figure 5: Histogram of bootstrap correlations

realisations of $(X, Y) \sim F_0$, which we estimate by the empirical distribution function. In Matlab, we proceed as follows

```
for b=1:10000
    i=ceil(15*rand(1,15));
    tstar(b)=corr(law(i,1),law(i,2));
end
```

a histogram of $t(y^*)$ is given in Figure ???. If we want to test the hypothesis that the two scores are independent, this can be done by constructing a confidence interval of $\text{Corr}(X, Y)$

```
Delta=tstar-0.776;
quantile(Delta,0.95)
```

```
áans=0.1706;
```

This gives a lower interval $[0.776 - 0.1706, 1] = [0.605, 1]$ that does not contain zero. Hence the hypothesis can be rejected.

Example 0.6 (Ph-levels). Figure ??? gives histograms of historical and current measurements of Ph levels at each of 149 lakes in Wisconsin. The data are stored in `ph.mat` as the vectors `ph1` and `ph2` respectively. Historical data from 25 of the lakes are missing, so paired sample comparisons are not possible. A reasonable model is that historical and current measurements are all independent and with distribution functions F_0 and G_0 respectively, hence we have parameter-space $\Theta = \mathcal{F} \times \mathcal{F}$, where \mathcal{F} is the set of distribution functions on

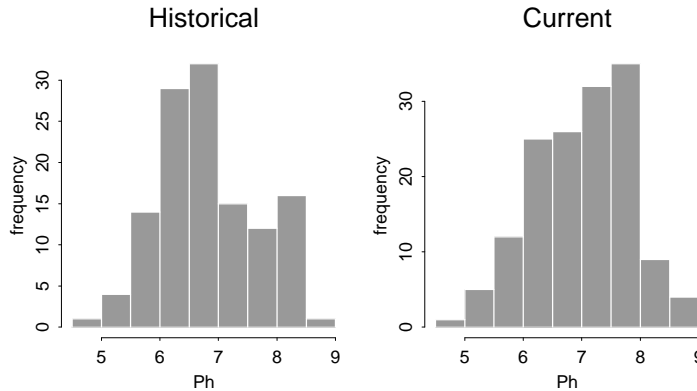


Figure 6: Histograms of historical and current Ph levels in Wisconsin lakes

\mathbf{R} and P_0 is the distribution with distribution function $\prod_{i=1}^{124} F_0(y_i) \prod_{i=125}^{273} G_0(y_i)$, where y_1, \dots, y_{124} and y_{125}, \dots, y_{273} correspond to the historical and current measurements respectively. It is of some interest to examine whether the Ph-levels have increased, and hence we compute the difference in medians of the historical and current populations, $t(y) = \hat{G}^{-1}(1/2) - \hat{F}^{-1}(1/2)$ which turns out to be 0.422. This suggests that either there has been an increase or that $t(y)$ has overestimated the true value $G_0^{-1}(1/2) - F_0^{-1}(1/2)$, to assess the latter we want to examine the distribution of $t(Y)$ which we approximate by the distribution of $t(Y^*)$ using the Bootstrap. Each bootstrap simulation consists of simulating from the empirical distribution function of the two separate samples, obtaining the median of each and differencing.

```

for b=1:10000
    ih=ceil(124*rand(1,124));
    ic=ceil(149*rand(1,149));
    tstar(b)=median(ph2(ic))-median(ph1(ih));
end

```

which gives the histogram of bootstrap differences in Figure ??, the fact that all simulated values exceed 0 give some extra support for the hypothesis that Ph-levels have increased. The histogram looks rather rugged, would you get a smoother histogram if you were bootstrapping the difference in means? Why?

The limitation of non-parametric models is that we need a reasonably large number of independent observations from each of the unknown distribution functions involved in the model in order to construct an accurate estimate of \hat{P} .

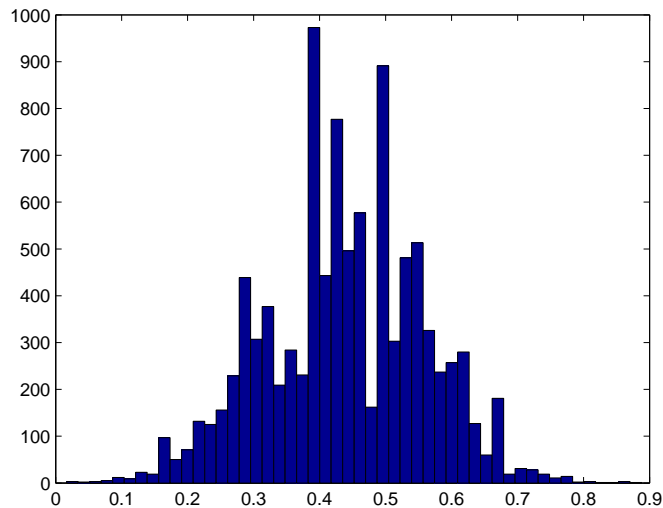


Figure 7: Histogram of simulated median differences

0.4 Bootstrapping parametric models

We have already seen an example the Bootstrap in parametric models in Example ??, here we will look at some more complicated models.

Example 0.7 (Space shuttle challenger). In 1986, the space shuttle Challenger exploded during take off, killing the seven astronauts aboard. It is believed that the explosion was caused by the failure of an *O-ring* (a rubber ring that seals parts of the ship together), and that the failure was caused by the cold weather at the time of launch (31° F). In the file `oring.mat`, you will find temperature (in Fahrenheit) and failure data from 23 shuttle launches, where 1 stands for O-ring failure and 0 no failure. See Figure ?. We want to model the failure-data as a function of temperature, a common model in this context is the *probit* regression model that asserts that observations y_i are independent Bernoulli($m(t_i)$), where t_i is temperature (which we will assume fixed) and $m(t) = \Phi(\beta_0 + \beta_1 t)$ (this is just a rather arbitrary way of mapping a linear regression to the unit interval). Hence, $\theta = (\beta_0, \beta_1) \in \mathbf{R}^2 = \Theta$. Further, given an estimate $\hat{\theta}$ it is straightforward to generate Bootstrap draws y^* from $P_{\hat{\theta}}$. The model is an example of a *generalised linear model* (GLM), a class of generalisations to the usual linear regression model with Normal errors. Matlab's `glmfit` can be used to estimate θ .

```
b=glmfit(chall(:,1),[chall(:,2) ones(23,1)],'binomial','probit');
```

if we are interested in the failure-probability at 65F, an estimate is given by

```
normcdf(b(1)+b(2)*65)
ans =
```

0.4975

and to assess the accuracy, we construct a 95% confidence interval using the parametric Bootstrap

```
for i=1:1000
    ystar=binornd(1,normcdf(b(1)+b(2)*chall(:,1)));
    bstar(i,1:2)=glmfit(chall(:,1),[ystar ones(23,1)],'binomial','probit');
end
tstar=normcdf(bstar(:,1)+bstar(:,2)*65);
Delta=tstar-.4975;
C=[.4975-quantile(Delta,.975) .4975-quantile(Delta,.025)]
C =

    0.0792    0.7484
```

which turns out quite wide.

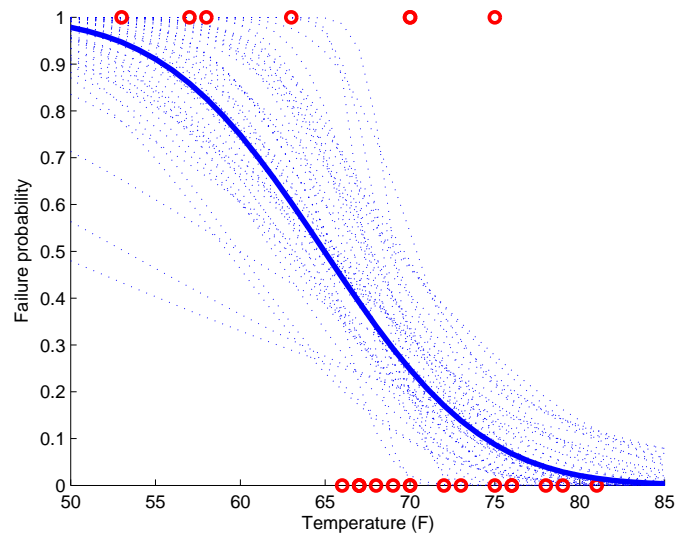


Figure 8: Data for the Challenger example (circles), fitted regression line (thick solid) and 50 draws from its Bootstrap distribution (dotted)

Example 0.8 (Hormone levels). The file `hormone.mat` contains a time-series of lutenizing hormone level in blood-samples. A simple model for time-series is the Gaussian first-order autoregressive (AR(1)) process defined recursively through

$$Y_i = X_i + \mu, \quad X_i = \alpha X_{i-1} + \epsilon_i, \quad (2)$$

where ϵ_i , $i = 1, \dots, n$ are independent $N(0, \sigma_2)$ and $X_0 \sim N(0, \sigma^2/(1 - \alpha^2))$, $|\alpha| < 1$. Under this specification, $Y = (Y_1, \dots, Y_n) \sim N((\mu, \dots, \mu), \Sigma)$, where Σ has elements $\Sigma_{ij} = \alpha^{|i-j|} \sigma^2 / (1 - \alpha^2)$ and $\theta = (\mu, \sigma^2, \alpha) \in \mathbf{R} \times \mathbf{R}^+ \times (-1, 1) = \Theta$. The parameters can be estimated by their Maximum-Likelihood estimators, but we choose a simpler approach and estimate μ by the empirical mean, α by the empirical first order autocorrelation and σ^2 by the empirical variance multiplied by $1 - \hat{\alpha}$:

```
mhat=mean(y);
ahat=corr(y(1:47),y(2:48));
s2hat=var(y)*(1-ahat);
```

\hat{P} is now $N(\hat{\mu}, \hat{\Sigma})$, and we can simulate from this distribution using the Choleski method (or Matlabs `mvnrnd` which is an implementation of the same). The covariance matrix $\hat{\Sigma}$ can be computed by

```
Shat=s2hat/(1-ahat)*ahat.^toeplitz(0:47,0:47);
```

If we are interested in the properties of our estimate of α , we proceed as follows

```
for b=1:1000
    ystar=mvnrnd(mhat*ones(1,48),Shat)';
    tstar(b)=corr(ystar(1:47),ystar(2:48));
end
```

In the right panel of Figure ?? we have plotted a histogram of T^* and in Figure ?? four realisations of the timeseries Y^* .

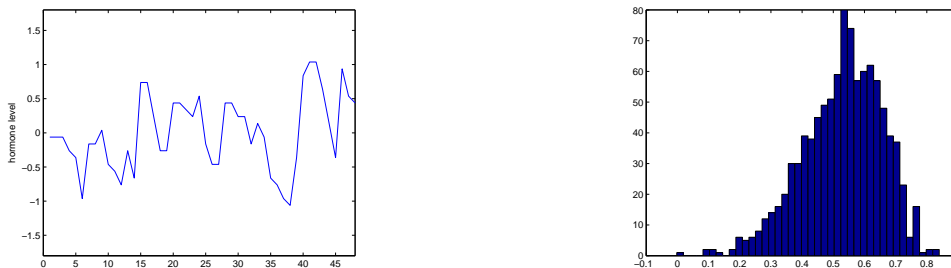


Figure 9: Left panel: Hormone level data-set. Right panel: Histogram of $T^* = \hat{\alpha}^*$.

0.5 Bootstrap for semi-parametric models

A semi-parametric model can be viewed as a non-parametric model with some additional parametric structure. This allows us to relieve some of the rather restrictive model assumptions in Section ??.

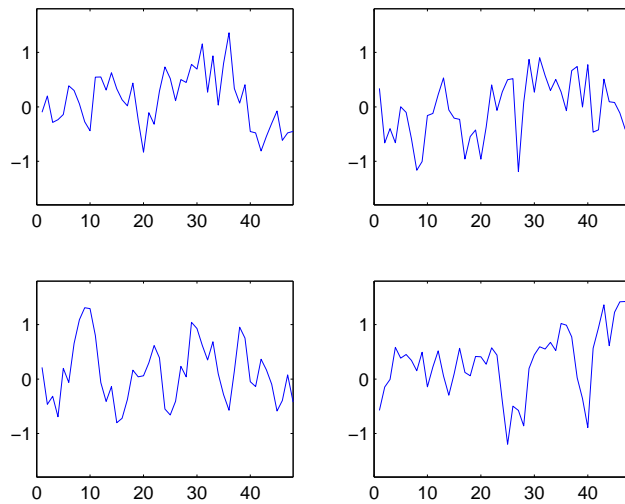


Figure 10: Four realisations of Y^* for the hormone data

0.5.1 Regression models

A regression model tries to establish a functional relationship between a set of responses $y = (y_1, \dots, y_n) \in \mathbf{R}^n$ and a set of covariates/design-variables $x = (x_1, \dots, x_n) \in \mathbf{R}^d \times \mathbf{R}^n$. There are two classes of such models, in a regression model with *fixed design* we view the covariates x as fixed or user-defined quantities. In a model with *random design*, both x and y are observations of random variables. An example of the first could be that we measure the daily temperature over a period of time, here y_i could be measured temperature on day i and $x_i = i$. An example of the second could be that (x_i, y_i) are measured weight and height of a randomly chosen individual, and we want to model weight as a function of height.

Fixed design linear regression

Consider data $y = (y_1, \dots, y_n)$ generated from the model

$$Y_i = \alpha + \beta x_i + \epsilon_i, i = 1, \dots, n, \quad (3)$$

where β and x_i might be a row and column vector respectively, the residuals ϵ_i are i.i.d. from F_0 and $E(\epsilon_i) = 0$. In the absence of further assumptions on the distribution of the residuals, this is a semi-parametric model with unknown parameter $\theta_0 = (\alpha_0, \beta_0, F_0)$. Moreover, P_0 is the distribution corresponding to distribution function

$$F(y) = \prod_{i=1}^n F_0(y_i - \alpha - \beta x_i).$$

How do we find an estimate \hat{P} of this distribution? After all, we actually observe the residuals ϵ_i and hence can't estimate F_0 by the empirical distribution function directly. The solution is to start with the parametric part, i.e. we first find estimates $(\hat{\alpha}, \hat{\beta})$ of α and β , e.g. by least-squares

$$(\hat{\alpha}, \hat{\beta}) = \operatorname{argmin}_{(\alpha, \beta)} \sum_{i=1}^n (y_i - \alpha - \beta x_i)^2. \quad (4)$$

Then we compute the *approximate residuals* $\hat{\epsilon}_i = y_i - \hat{\alpha} - \hat{\beta}x_i$. These will be approximately independent realisations from F_0 and hence we can estimate the distribution by

$$\hat{F}(u) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{\hat{\epsilon}_i \leq u\},$$

the empirical distribution of the approximate residuals. Sampling Y^* now proceeds as follows

1. Draw i_1, i_2, \dots, i_n independently from the uniform distribution on the set of integers $\{1, 2, \dots, n\}$.
2. $y^* = (y_1^*, \dots, y_n^*) = (\hat{\alpha} + \hat{\beta}x_{i_1} + \hat{\epsilon}_{i_1}, \dots, \hat{\alpha} + \hat{\beta}x_{i_n} + \hat{\epsilon}_{i_n})$ is now a draw from Y^* with distribution $\prod_{i=1}^n \hat{F}(y_i - \hat{\alpha} - \hat{\beta}x_i)$.

Example 0.9. We illustrate the above residual Bootstrap on a simulated data-set following the slightly more complicated model

$$Y_i = \alpha + \beta x_i + x_i \epsilon_i, i = 1, \dots, n, \quad (5)$$

where the ϵ_i are independent from F_0 , i.e. the size of the “measurement errors” is proportional to x_i . Data is plotted in Figure ??.

In order to estimate F_0 , we need to construct approximate residuals from this distribution. First we estimate α and β using least-squares (though a weighted version would have been more appropriate in this case). Matlab gives

```
polyfit(x,y,1)
```

```
ans =
```

```
5.2794    1.2027
```

where the first is slope β and second intercept α . Residuals are now given by $\hat{\epsilon}_i = (y_i - \hat{\alpha} - \hat{\beta}x_i)/x_i$, in Matlab

```
ehat=(y-1.2027-5.2794*x)./x;
```

and Bootstrap proceeds by

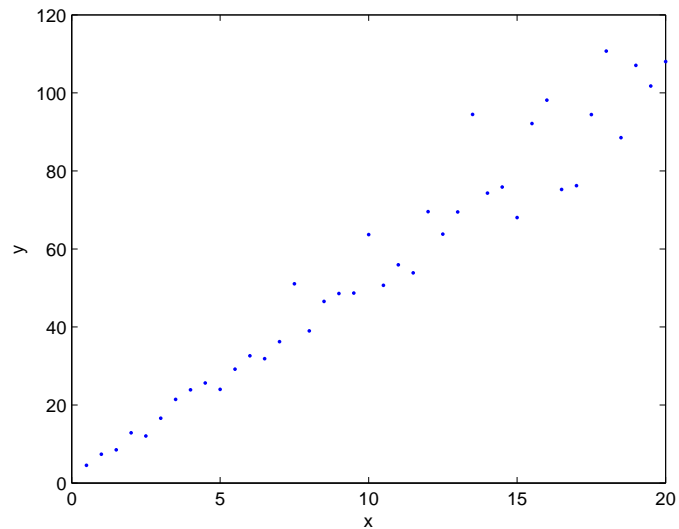


Figure 11: Data from the model in ??

```

for b=1:1000
    i=ceil(40*rand(1,40));
    estar=ehat(i);
    ystar=1.2027+5.2794*x+estar.*x;
    tstar(b,1:2)=polyfit(x,ystar,1);
end

```

In Figure ?? we have plotted 20 draws from the regression line.

Random design linear regression

In random design regression, $(x_i, y_i), i = 1, \dots, n$, are independent draws from a bivariate distribution $F_0(x, y)$, with the property

$$E(Y|X = x) = \alpha + \beta x. \quad (6)$$

Bootstrap from this model is equivalent to that of Example ??, i.e. with \hat{P} corresponding to the distribution $\prod_{i=1}^n \hat{F}(x_i, y_i)$, where \hat{F} is the (bivariate) empirical distribution function..

Example 0.10 (Bodies and brains). The left panel of Figure ?? shows average brain weight (g) against body weight (kg) for 62 species of mammals, in the right panel the same data is plotted on logarithmic scales, showing an approximate linear relation. Hence, with y denoting log-brain weight and x log-body weight we might use the model in (??). Bootstrapping the regression lines proceeds as follows (c.f. Example ??)

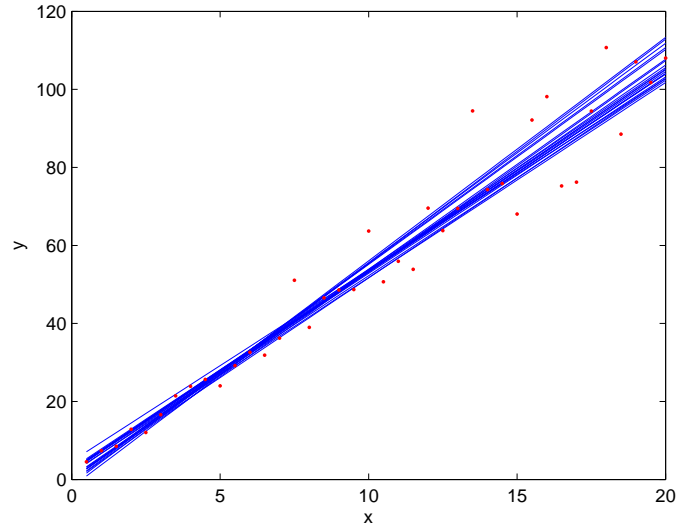


Figure 12: Data and regression lines estimated from each of 20 draws from Y^* .

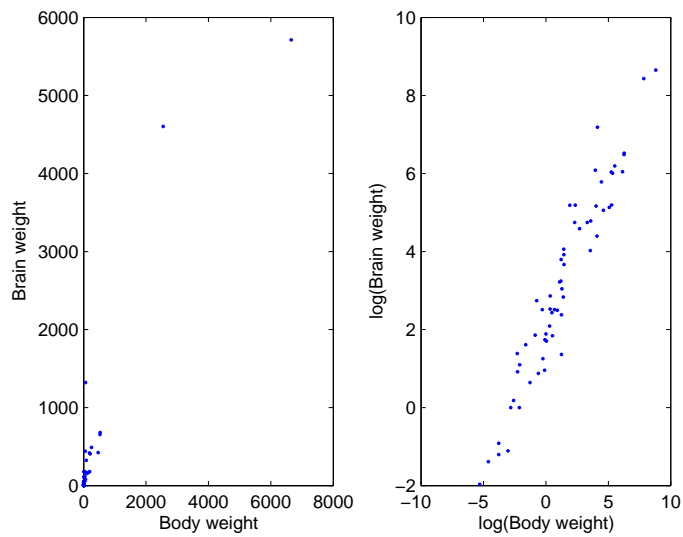


Figure 13: Average body and brain weights for 62 mammals.

```
x=log(mammals(:,1)); y=log(mammals(:,2));
for b=1:1000
    i=ceil(62*rand(1,62));
    ystar=y(i);
    xstar=x(i);
    tstar(b,1:2)=polyfit(xstar,ystar,1);
end
```

Figure ?? shows a few regression lines estimated from draws from (X^*, Y^*) .

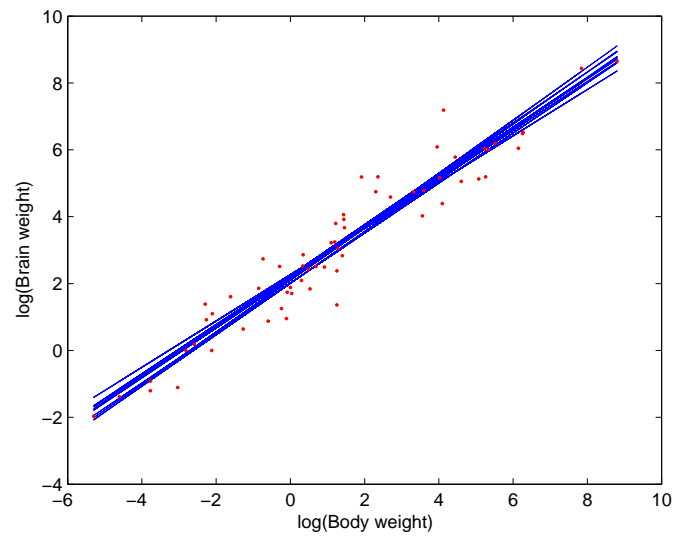


Figure 14: Bootstrap regression lines for the mammals data.