Computational Statistics - Suggested Solution for Exam

Frank Miller, IDA, Linköping University

2025-08-29

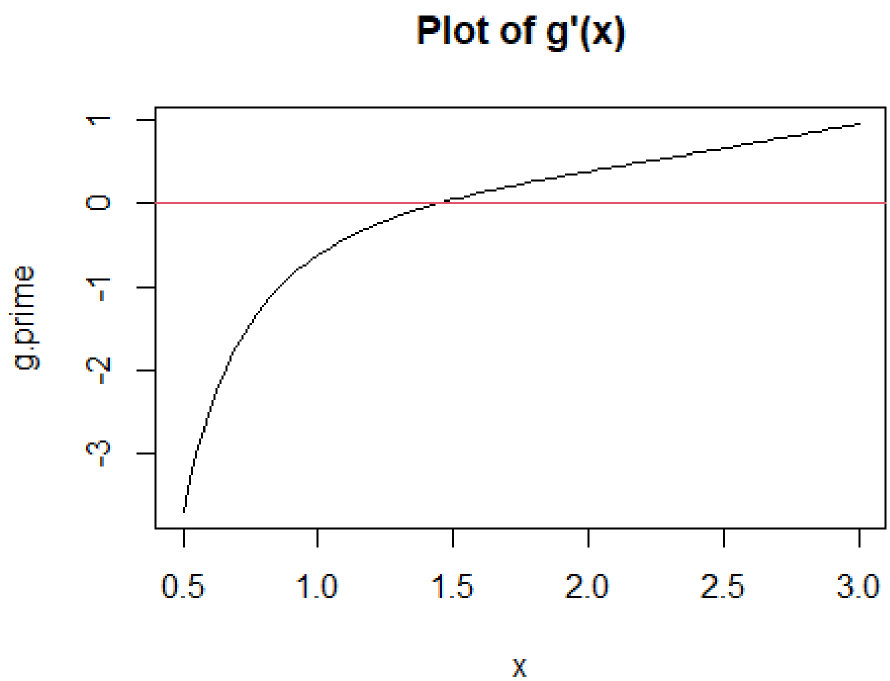
Question 1

Question 1a

Plots of g(x) and g.prime(x)

```
# Define function g and g'
g       <- function(x) { 2^x/x }
g.prime <- function(x) { (x*log(2)-1)*2^x/x^2 }

# plots
plot(g, xlim=c(0.5,3), main="Plot of g(x)")
```


```
plot(g.prime,xlim=c(0.5,3), main="Plot of g'(x)")
abline(h=0, col="2")
```



Based on the plots, the minimum should be attained in the neighborhood of 1.5.

Question 1b

Secant method

```
secant <- function(f=g.prime, x0, x1, tol=0.0001){
  conv <- abs((g.prime(x1)-g.prime(x0)))
  iter <- 1
  while(conv>tol){
    x2   <- x1-g.prime(x1)*(x1-x0)/(g.prime(x1)-g.prime(x0))
    conv <- abs((g.prime(x1)-g.prime(x0)))
    x0   <- x1
    x1   <- x2
    iter <- iter+1
  }
  return(list(x2, g(x2), iter))
}

res1 <- secant(g.prime, 1, 1.5)
res1

## [[1]]
## [1] 1.442695
##
## [[2]]
## [1] 1.884169
```

```
##
## [[3]]
## [1] 6
```

The minimum is attained at $x = 1.442695$ with $f(x) = 1.884169$. The number of iterations until the convergence of the algorithm is 6.

Question 1c

With optimize:

```
res2 <- optimize(g, c(1, 2), tol = 0.0001)
```

The minimum is attained at $x = 1.442691$ with $f(x) = 1.884169$. The result obtaine by optimize is almost the same as obtained by the secant method. The difference is only in the 6th decimal.

Algebraical calculation

$x(\log(2) - 1) * 2^x/x^2 = 0.$
Multiply the both sides by x^2: $x(\log(2) - 1) * 2^x = 0.$
Divide both sides by 2^x: $x(\log(2) - 1) = 0.$
Therefore: $x * \log(2) = 1$ and $x = 1/\log(2).$

```
res3 <- 1/log(2)
```

The minimum is attained at $x = 1.442695$, i.e., the secant method has identified the correct solution up to these 6 digits.

Question 2

Question 2a

Load first the R-dataset using the R-function load.

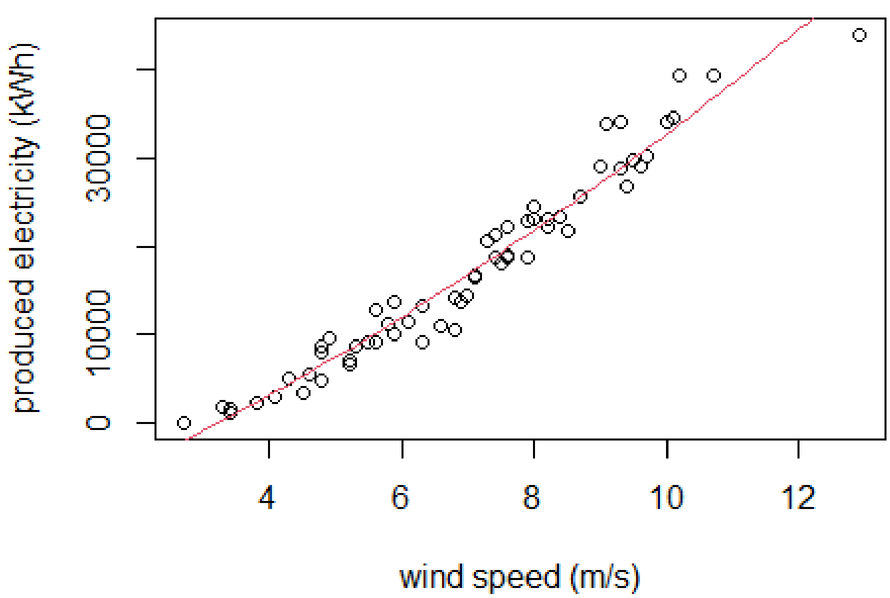
```
plot(data, type="p", xlab="wind speed (m/s)", ylab="produced electricity
(kWh)")
speed  <- data[, 1]
speed2 <- speed^2
power  <- data[, 2]
fit <- lm(power~speed+speed2)
summary(fit)

##
## Call:
## lm(formula = power ~ speed + speed2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6265.0 -1302.6   -53.7  1522.6  6268.5
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11486.82    2697.37  -4.259 7.63e-05 ***
## speed         3165.29     776.43   4.077 0.000141 ***
## speed2         125.47      53.41   2.349 0.022231 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2397 on 58 degrees of freedom
## Multiple R-squared:  0.9531, Adjusted R-squared:  0.9515
## F-statistic: 588.9 on 2 and 58 DF,  p-value: < 2.2e-16
```

```r
confint(fit)
```

```
##                   2.5 %     97.5 %
## (Intercept) -16886.20333 -6087.4465
## speed         1611.09096  4719.4818
## speed2          18.56761   232.3741
```

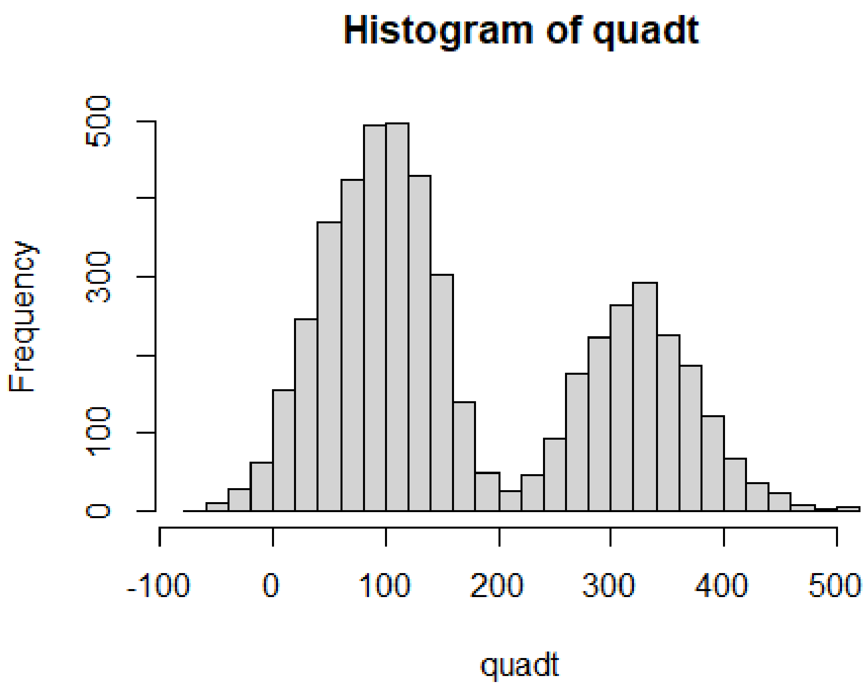
```r
betahat <- summary(fit)$coef[,1]
speedplot <- seq(2, 15, by=0.1)
lines(speedplot, betahat[1]+betahat[2]*speedplot+betahat[3]*speedplot^2,
col=2)
```



The 95%-confidence interval for the quadratic term based on assumption of normally distributed data is [18.6,232.4].

Question 2b

```r
# bootstrap CI for slope
set.seed(2025)
B <- 5000
quadt <- NULL
for (i in 1:B){
  ind <- sample(1:length(speed), replace=TRUE)
  s0  <- speed[ind]
  s02 <- s0^2
  p0  <- power[ind]
  quadt <- c(quadt, summary(lm(p0~s0+s02))$coef[3, 1])
}
hist(quadt, breaks=40)
```




```r
sqt <- sort(quadt)
round(c(sqt[round(0.025*B)], sqt[round(0.975*B)]), 3)

## [1]   3.485 404.675
```

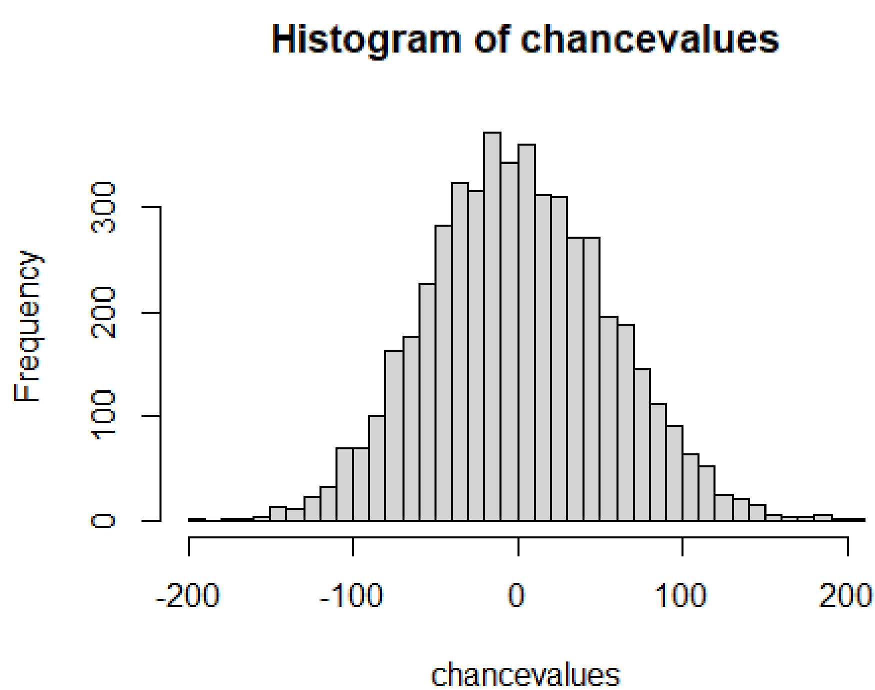
The 95%-bootstrap confidence interval is larger than the interval based on normal distribution assumption, especially at the upper bound. Especially the single value at a large wind speed has influence: If this value is not included into the bootstrap sample, the quadratic coefficient is larger (and is the reason for this bimodal bootstrap sample).

Question 2c

Note: A correct solution is here quite advanced. The important thing is here the general permutation test idea. If you show that, you will obtain points and can also obtain full points even if you not have all details in following solution:

Under the null hypothesis, we have a linear regression model. To apply a permutation test, we generate first residuals under the linear regression model (reduced model). We permute then the residuals, i.e., we assign randomly the residuals to each wind speed value. We add the permuted residuals to the predicted values for the reduced model.

```
fitlin <- lm(power~speed)
powpred <- predict(fitlin)
powresi <- resid(fitlin)
# Permutation test
chancevalues <- NULL
for (i in 1:B){
  ind <- sample(1:length(speed))
  t0  <- powpred + powresi[ind]
  chancevalues <- c(chancevalues, summary(lm(t0~speed+speed2))$coef["speed2",
"Estimate"])
}
hist(chancevalues, breaks=40)
```




```
pval <- mean(chancevalues>betahat[3])
pval
```

```
## [1] 0.0132
```

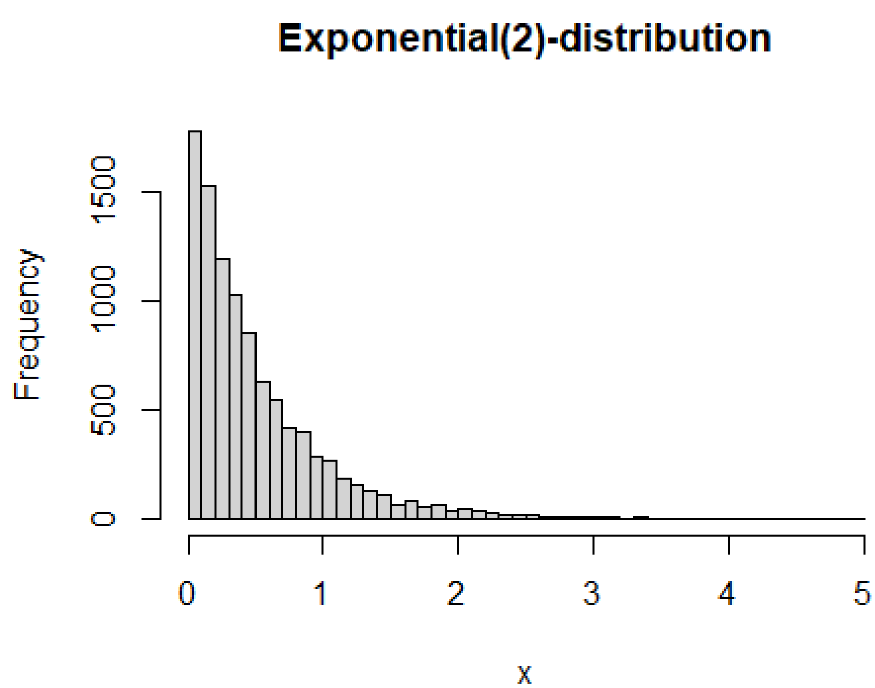
The p-value is 0.013 and we can reject the null hypothesis that $\beta_2 = 0$.

Question 3

Question 3a

One needs the cumulative distribution function F and then inverting it. Either, it can be found in the math-lecture 2 in the available material, or one could do the derivation directly as well: $F(x) = 1 - \exp(-\lambda x), x \geq 0$, and $y = 1 - \exp(-\lambda x)$ giving $F^{-1}(y) = -\log(1-y)/\lambda, 0 \leq y \leq 1$.

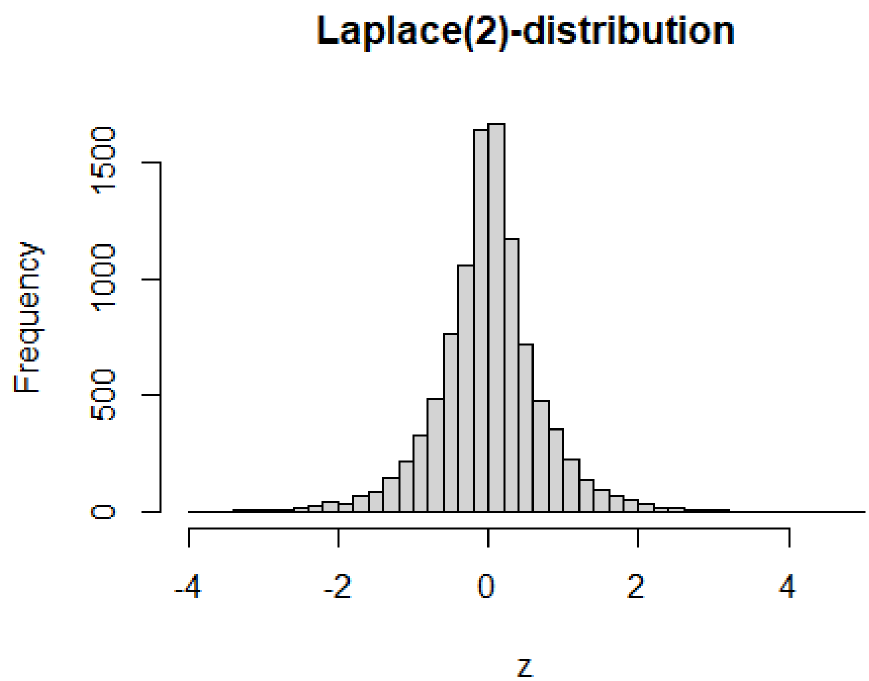
```
set.seed(2025)
y <- runif(10000)
x <- -log(1-y)/2
hist(x, breaks=50, main="Exponential(2)-distribution")
```



Question 3b

We use the generated values in part a and multiply them with either -1 or 1 (with 50% probability each, we generate it based on a uniform distribution, but using the function sample is fine, to), which is a kind of composition sampling.

```
zsign <- 2*rbinom(10000, size=1, prob=0.5) - 1
z     <- x*zsign
hist(z, breaks=50, main="Laplace(2)-distribution")
```



Question 3c

```
mean(z)

## [1] 0.007941876

var(z)

## [1] 0.5060617

mean(z<0.5)

## [1] 0.817
```

Based on the sample, the mean is 0.008 (theoretically, it should be 0 since the Laplace distribution is symmetric around 0). The variance is around 0.506, the probability that a sample is < 0.5 is around 0.82.