Computational Statistics - Suggested Solution for Exam

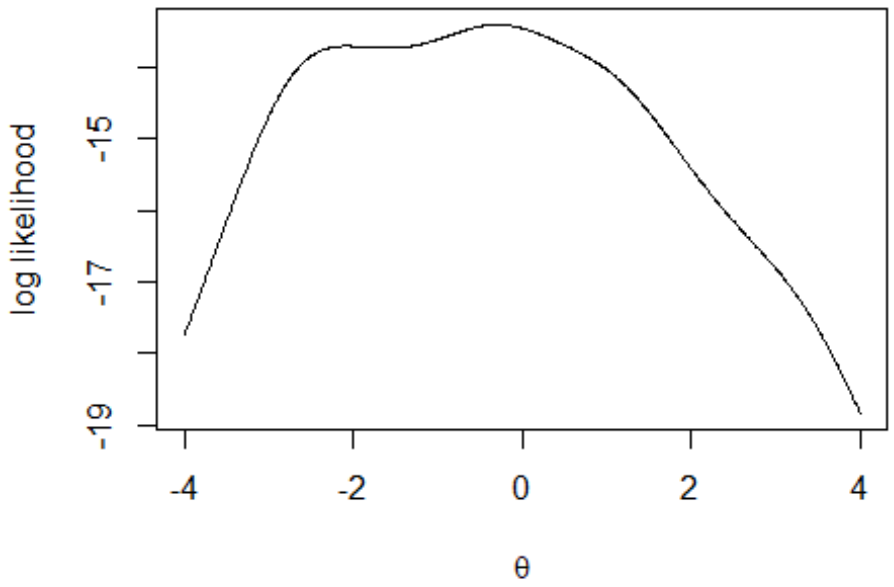
Frank Miller, IDA, Linköping University

2025-03-24

Question 1

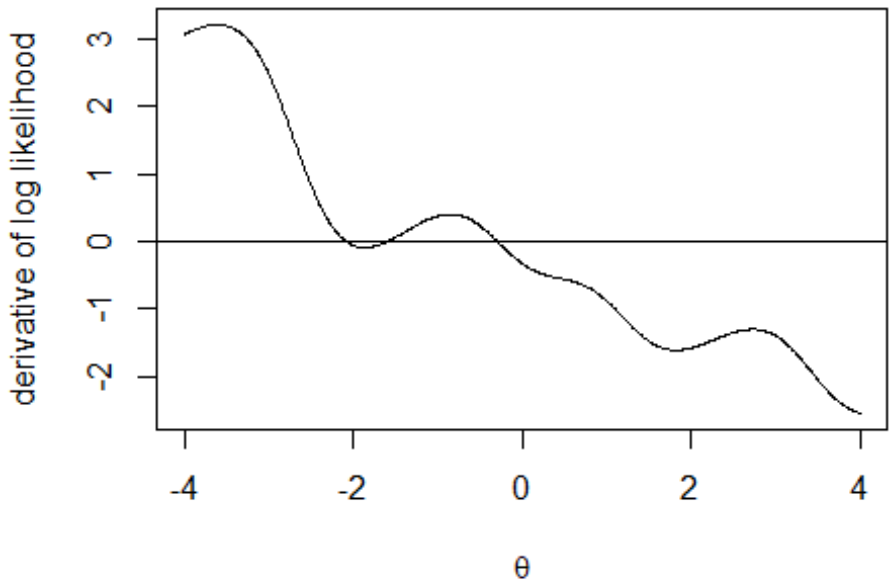
Question 1a

```
cauchyloglik <- function(theta, x){
  res <- c()
  for (t in theta)
    res <- c(res, -length(x)*log(pi)-sum(log(1+(x-t)^2)))
  res
}
tv  <- -400:400/100
obs <- c(-2.8, 3.4, 1.2, -0.3, -2.6)
plot(tv, cauchyloglik(tv, obs), type="l", xlab=expression(theta), ylab="log
likelihood")
```




```
cauchyderiv <- function(theta, x){
  res <- c()
  for (t in theta)
  res <- c(res, sum(2*(x-t)/(1+(x-t)^2)))
```

```
  res
}

plot(tv, cauchyderiv(tv, obs), type="l", xlab=expression(theta),
ylab="derivative of log likelihood")
abline(h=0)
```



The derivative is three times equal to 0. The first and the third time, it corresponds to a local maximum (since the derivative is decreasing from being positive to negative). The second root corresponds to a local minimum.

Question 1b - Newton

We use here the Newton method, which should be both fast in running time and straightforward to program, once we have determined the second derivative. We can derive the second derivative algebraically or use the D function in R as help:

```
D(expression(2*(x-t)/(1+(x-t)^2)), "t")

## -(2/(1 + (x - t)^2) - 2 * (x - t) * (2 * (x - t))/(1 + (x - t)^2)^2)

cauchyder2 <- function(theta, x){
  res <- c()
  for (t in theta)
    res <- c(res, sum(-2/(1+(x-t)^2) + 4*(x-t)^2/(1+(x-t)^2)^2))
  res
}
```

```
newton <- function(start, eps, x){
  told <- start - 9
  t <- start
  while (abs(t-told)>eps){
    told <- t
    t <- t - cauchyderiv(t, x)/cauchyder2(t, x)
  }
  t
}
```

We choose the following two starting values: 0 (slightly larger than the last local maximum according to plot) and -2.5 (slightly lower than the first local maximum). However, we need to check that we do not pick the local minimum.

```
t1 <- newton(start=0, eps=0.0001, x=obs)
t1
```

```
## [1] -0.2952455
```

```
cauchyloglik(t1, obs)
```

```
## [1] -13.40942
```

```
cauchyder2(t1, obs)
```

```
## [1] -1.230144
```

```
t2 <- newton(-2, 0.0001, obs)
t2
```

```
## [1] -2.082124
```

```
cauchyloglik(t2, obs)
```

```
## [1] -13.7077
```

```
cauchyder2(t2, obs)
```

```
## [1] -0.8813053
```

We have found the two local minima, since the second derivative at the identified positions is negative. Since the log likelihood is larger at t_1, $\theta = -0.2952$ is the global maximum (i.e., it is the Maximum Likelihood estimate for θ).

Problem 1c

Convergence speed: In general, Newton is fastest, secant is still fast but a little less than Newton, bisection is in general slow. Sensitivity to starting values: Newton is in general very sensitive, secant as well, bisection is not as sensitive and finds always some point with $f' = 0$ if the starting interval $[a, b]$ has property $f'(a)f'(b) < 0$. Bisection and secant just need the first derivative, Newton needs even second. Programming effort: If the second

derivative is provided, the programming effort is similar, slightly higher for bisection. If the second derivative needs to be derived, secant requires the lowest programming effort.

Problem 2

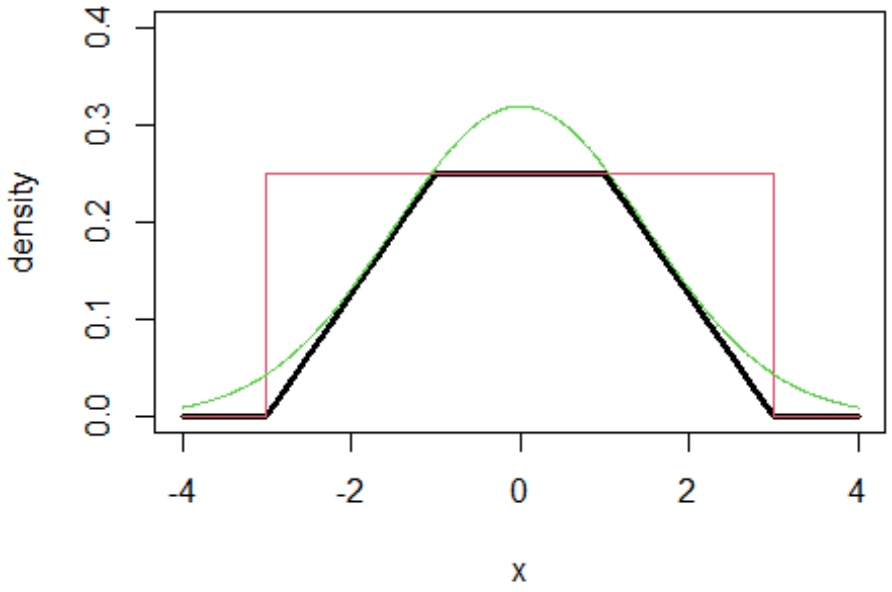
Question 2a

We can use the uniform distribution on the interval $[-3,3]$ as e_1, since $f > 0$ only on this interval. The density is equal to $1/6$ on this interval for that uniform distribution. Since f is at most $1/4$, we need to divide the density by $a_1 = 2/3$ to get an envelope.

f is symmetric around 0; therefore can we choose a normal density with mean 0 for e_2. Given the interval where f is positive, a standard deviation of 1 or a bit larger will work; we choose here sd=1.5. The scaling factor a_2 will be determined below by trial and error such that the plot shows the envelope property $e_2 \geq f$.

Question 2b

```
f <- function(x){
  (x>= -3)*(x<= -1)*(3+x)/8 + (x> -1)*(x<= 1)/4 +(x> 1)*(x<= 3)*(3-x)/8
}
e1 <- function(x){
  a1 <- 2/3
  ((x>= -3)*(x<= 3)/6)/a1
}
e2 <- function(x){
  a2 <- 5/6
  dnorm(x, mean=0, sd=1.5)/a2
}

xv <- -400:400/100
plot(c(-4,4), c(0,0.4), type="n", xlab="x", ylab="density")
lines(xv, f(xv), lwd=3)
lines(xv, e1(xv), col=2)
lines(xv, e2(xv), col=3)
```




```
min(e2(xv)-f(xv))
```

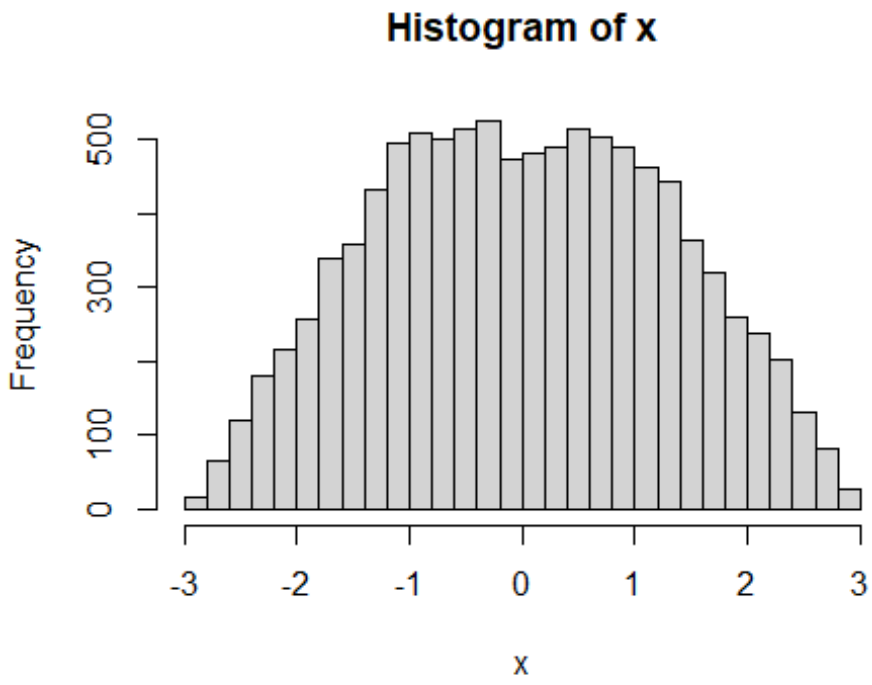
```
## [1] 0.005340386
```

e_1 and e_2 are larger than f on $[-4,4]$.

The envelope based on the normal distribution is here better since it yields to less waste. $a_2 = 5/6$ was sufficient; this means waste= 1/6. For e_1, the waste is 1/3 ($a_1 = 2/3$).

Question 2c

```
x <- c()
n <- 10000
while (length(x)<n){
  xcand <- rnorm(1, mean=0, sd=1.5)
  u <- runif(1, 0, 1)
  if (u<f(xcand)/e2(xcand)) x <- c(x, xcand)
}
hist(x, breaks=40)
```




```r
sd(x)
```

```
## [1] 1.291602
```

The histogram looks good compared to the density f. The standard deviation of X is approximately 1.29.

The following was not required for the question: The following code avoids loops and is much faster (works even for n=1 000 000 in less than 1s); It is assuming that at most 20% waste is generated (expected 16.7%):

```r
nsim <- round(n*1.25)
xcand <- rnorm(nsim, mean=0, sd=1.5)
uvect <- runif(nsim, 0, 1)
x <- xcand[(uvect<f(xcand)/e2(xcand))]
length(x)
```

```
## [1] 10490
```

```r
x <- x[1:n]
```

Question 3

Question 3a

```r
temp <- read.csv("tempLink.csv")
plot(temp$year, temp$tempav, type="p", xlab="year", ylab="Average temperature
(degrees Celcius)")
```

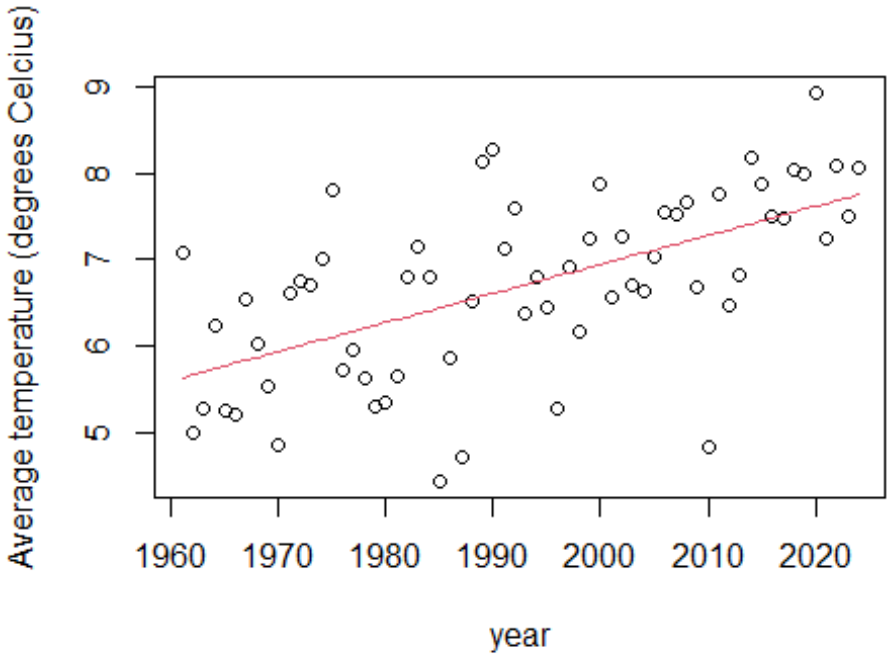
```r
fit <- lm(tempav~year, data=temp)
summary(fit)
```

```
##
## Call:
## lm(formula = tempav ~ year, data = temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.45315 -0.53133  0.04308  0.49201  1.71346
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -60.34534   11.43670  -5.276 1.78e-06 ***
## year          0.03365    0.00574   5.862 1.90e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8482 on 62 degrees of freedom
## Multiple R-squared:  0.3566, Adjusted R-squared:  0.3462
## F-statistic: 34.36 on 1 and 62 DF,  p-value: 1.902e-07
```

```r
confint(fit)
```

```
##                    2.5 %       97.5 %
## (Intercept) -83.20698582 -37.48370237
## year          0.02217266   0.04511937
```

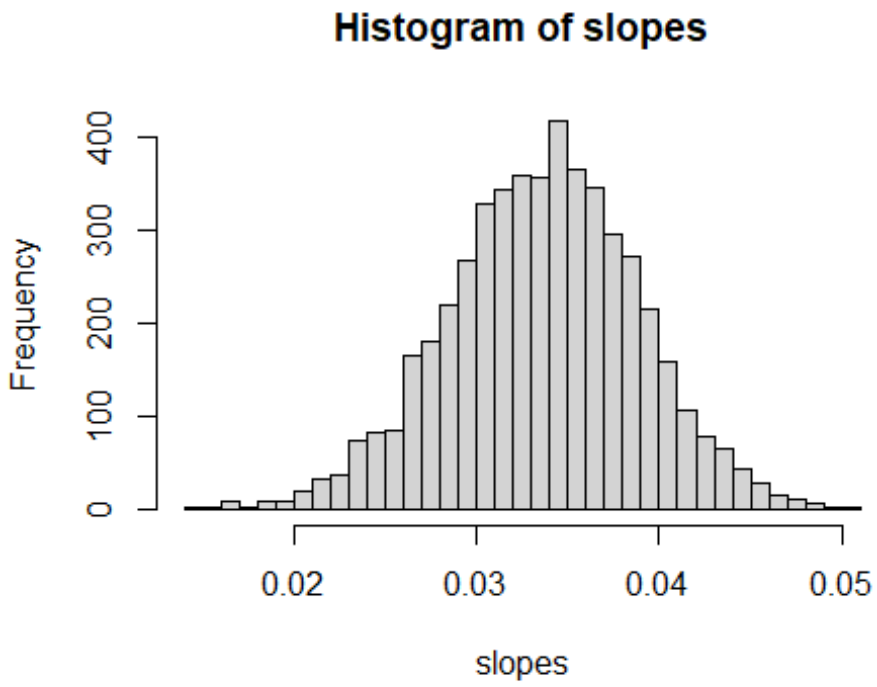
```r
betahat <- summary(fit)$coef[,1]
lines(temp$year, betahat[1]+betahat[2]*temp$year, col=2)
```



The 95%-confidence interval for the yearly increase in temperature based on assumption of normally distributed data is $[0.022, 0.045]$.

Question 3b

```
# bootstrap CI for slope
B <- 5000
slopes <- NULL
for (i in 1:B){
  ind <- sample(1:length(temp$year), replace=TRUE)
  y0  <- temp$year[ind]
  t0  <- temp$tempav[ind]
  slopes <- c(slopes, summary(lm(t0~y0))$coef[2, 1])
}
hist(slopes, breaks=40)
```



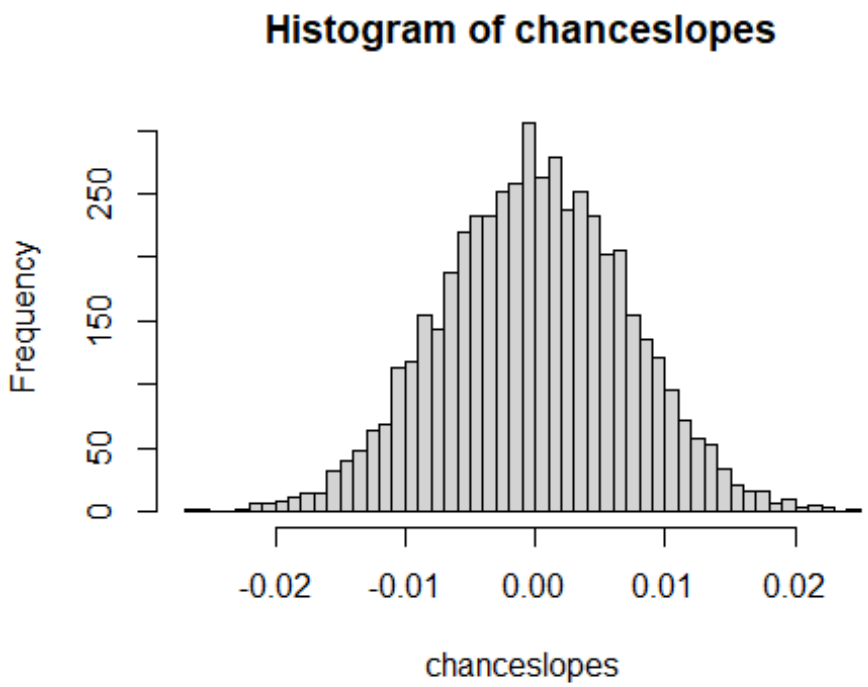

```
ssl <- sort(slopes)
round(c(ssl[round(0.025*B)], ssl[round(0.975*B)]), 3)

## [1] 0.023 0.044
```

The 95%-bootstrap confidence interval is similar to the interval based on normal distribution assumption. Given that we obtain almost the same CI whether or not we assume normally distributed data, we can rely on this CI. The increase of temperature per year is between around 0.022 and 0.045 degrees of Celcius (assuming a straight line regression model).

Question 3c

In contrast to the bootstrap, we do not resample pairs. We resample only temperatures and keep the years such that each year gets a temperature from an arbitrary year.

```
# Permutation test
chanceslopes <- NULL
for (i in 1:B){
  ind <- sample(1:length(temp$year))
  t0  <- temp$tempav[ind]
  chanceslopes <- c(chanceslopes, summary(lm(t0~temp$year))$coef[2, 1])
}
hist(chanceslopes, breaks=40)
```



Since the whole null-distribution generated here has values below the observed slope ($\hat{\beta}_1 = 0.0337$), the p-value is formally 0 for this run. Even if we would conservatively assume that in another run, one value of 5000 would be larger than $\hat{\beta}_1$, the p-value would be very low. The p-value of the permutation test would then be $1/B = 1/5000 = 0.0002$. Therefore, we can be sure that the p-value is < 0.001.