Day11 exercise solutions

Ali Movasati, Isabelle Caroline Rose Cretton, Tristan Koning

Nov. 15th, 2024

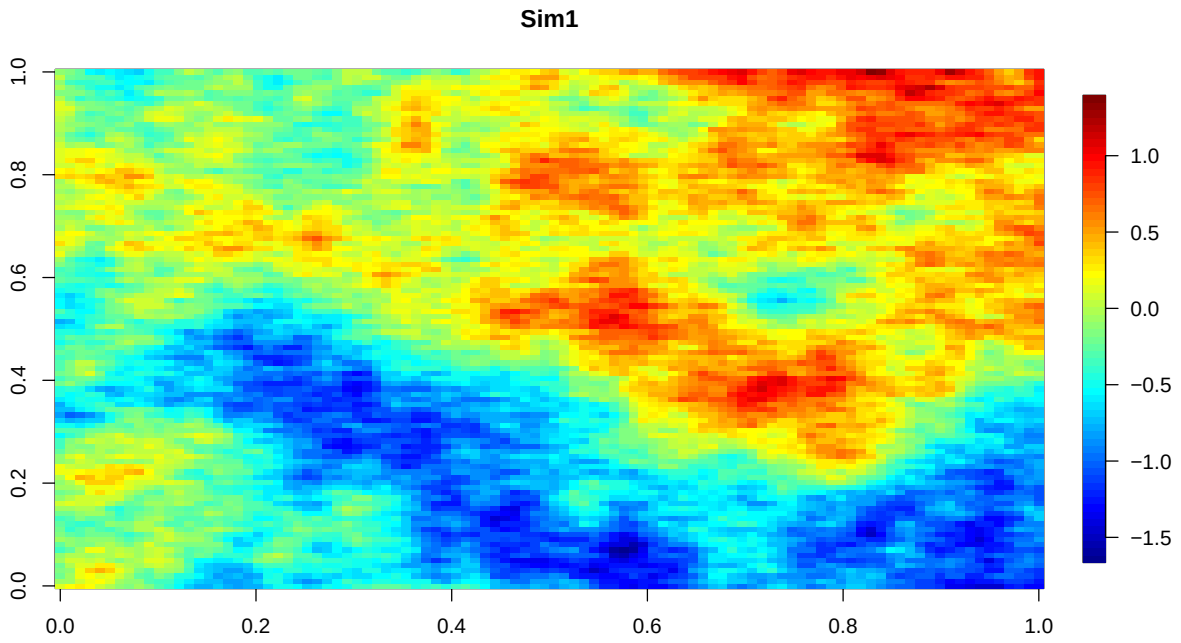
```
# Set global code chunk options
knitr::opts_chunk$set(
  echo = TRUE,
  warning = FALSE,
  message = FALSE,
  fig.width = 10,
  fig.height = 6
)
```

```
# load required libraries
library("fields")
library("skimr")
library("dplyr")
library("magrittr")
library("ggplot2")
library("survival")
library("survminer")
library("gridExtra")

# define functions
`%notin%` <- Negate(`%in%`)
```

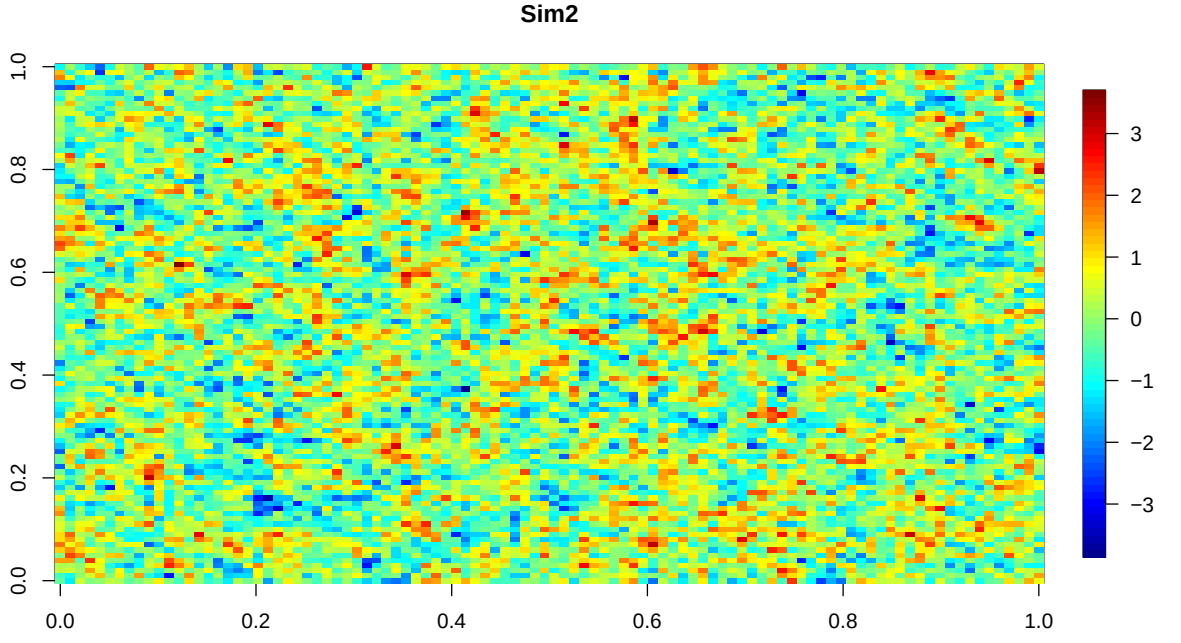
Problem 1

(a)

```
# Load day11/data/spatialSim.RData
load("data/spatialSim.RData")

# Plot the data
image.plot(sim1, main = "Sim1")
```




```
image.plot(sim2, main = "Sim2")
```




```
summary(as.vector(sim1))
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
```

```
## -1.64094 -0.55859 -0.05631 -0.13214  0.28576  1.37183
```

```
summary(as.vector(sim2))
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.80355 -0.63321  0.01862  0.01869  0.68658  3.64528
```

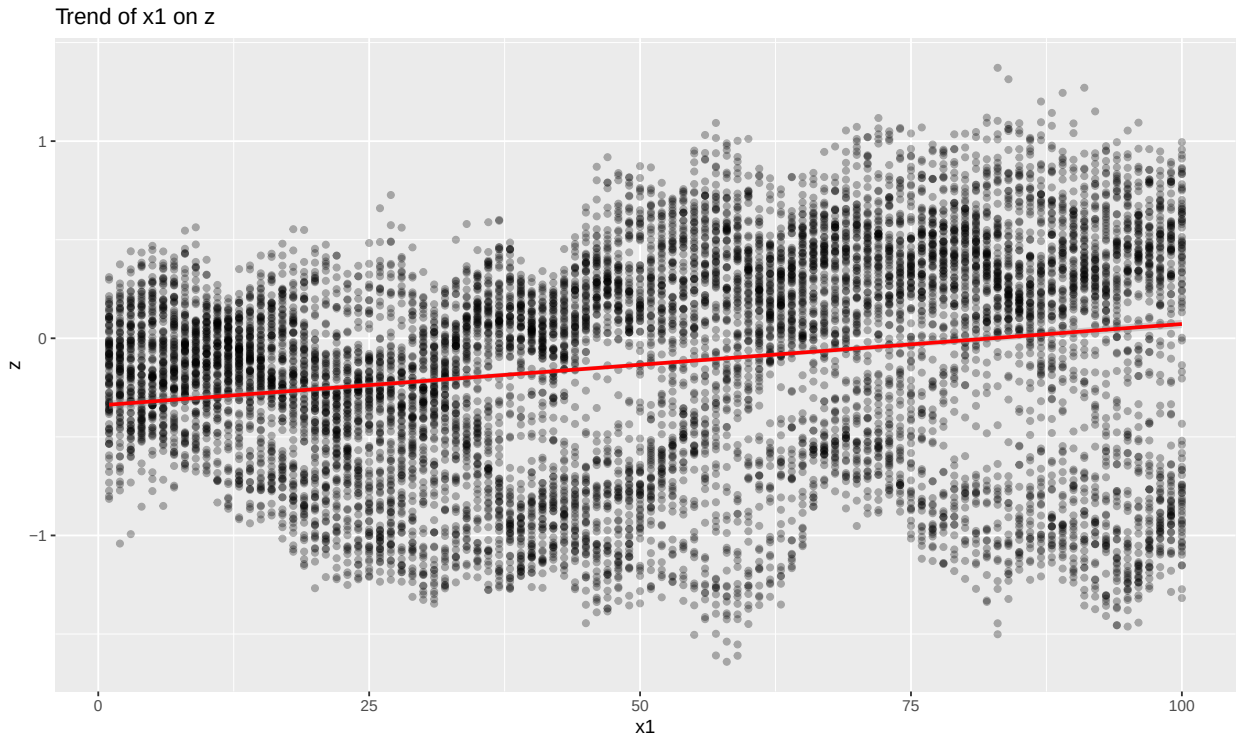
Looking at the plot, sim2 seems to be some sort of noise component, which varies more strongly than sim1 (values from -3.8 to 3.6). Sim1 seems to be more structured.

(c)

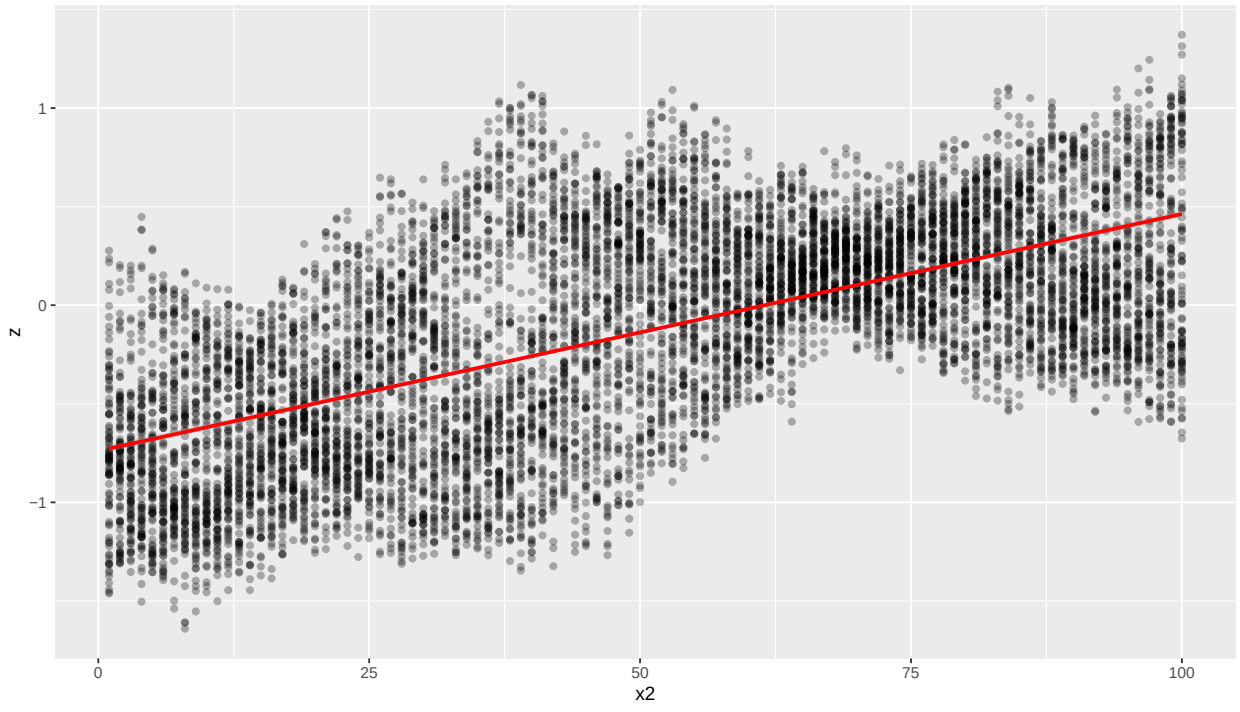
```
coords <- expand.grid(x1 = 1:nrow(sim1), x2 = 1:ncol(sim1))
sim1_df <- data.frame(coords, z = as.vector(sim1))

sim1_lm <- lm(z ~ x1 + x2, data = sim1_df)
summary(sim1_lm)
```

```
##
## Call:
## lm(formula = z ~ x1 + x2, data = sim1_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02896 -0.30692  0.00469  0.27550  1.32699
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.947859   0.011015  -86.06   <2e-16 ***
## x1           0.004133   0.000143   28.91   <2e-16 ***
## x2           0.012020   0.000143   84.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4128 on 9997 degrees of freedom
## Multiple R-squared:  0.4415, Adjusted R-squared:  0.4414
## F-statistic:  3951 on 2 and 9997 DF,  p-value: < 2.2e-16
```

```
ggplot(sim1_df, aes(x = x1, y = z)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Trend of x1 on z", x = "x1", y = "z")
```



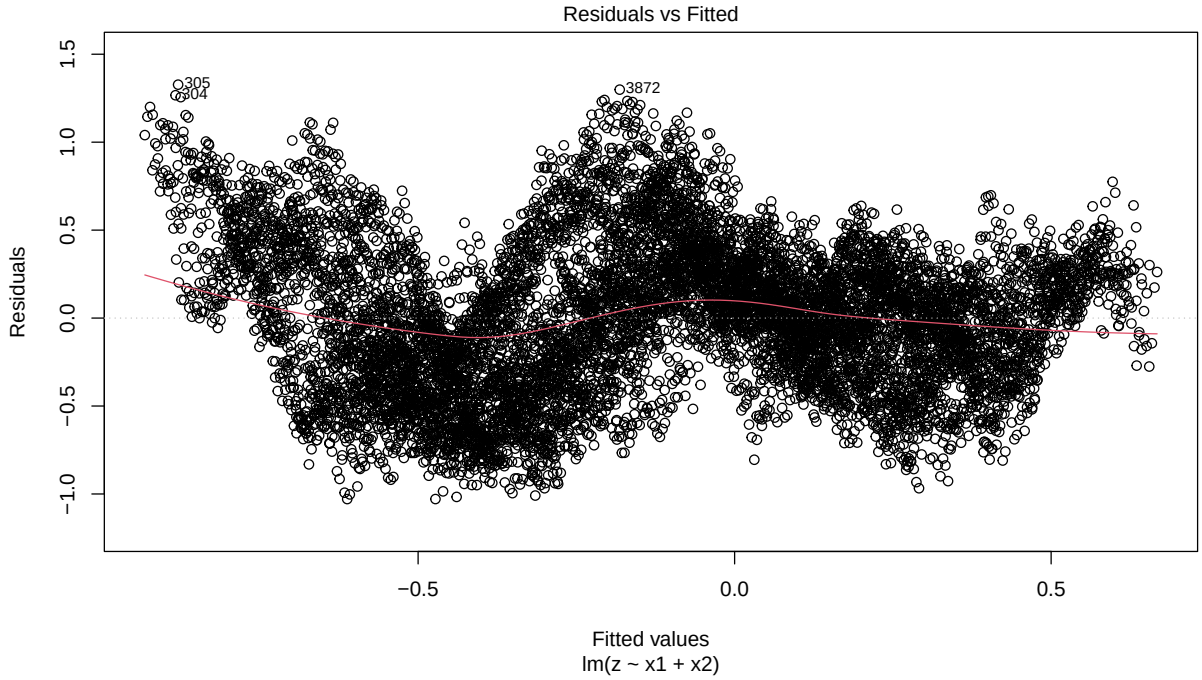

```
ggplot(sim1_df, aes(x = x2, y = z)) +
  geom_point(alpha = 0.3) +
  geom_smooth(method = "lm", col = "red") +
  labs(title = "Trend of x2 on z", x = "x2", y = "z")
```

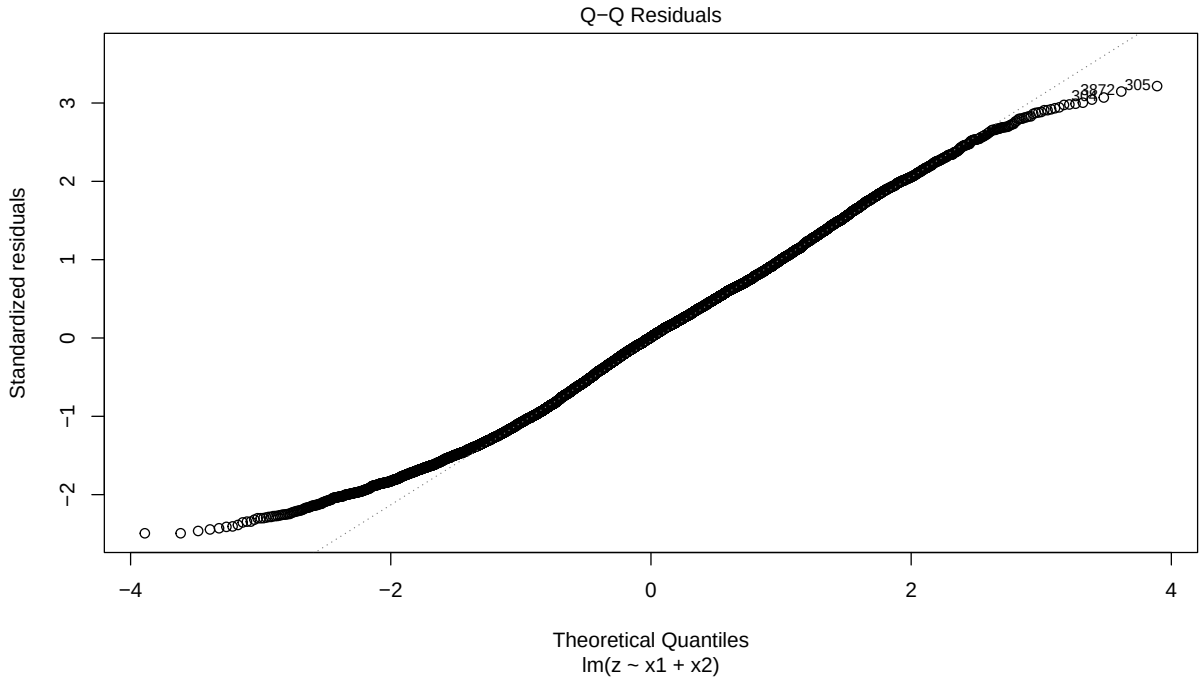


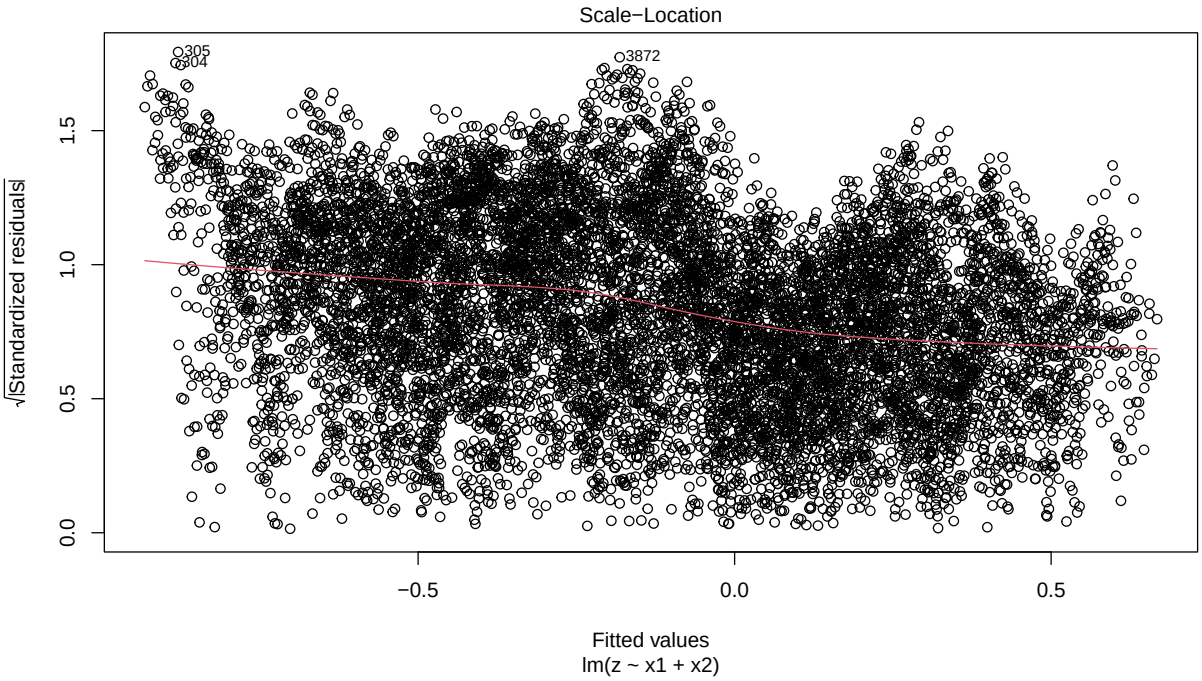
There appears to be a trend for both x1 and x2 that the higher these coordinate values go, so does the value of z. This is more pronounced on z. This makes sense as we seem to have a hotspot in the "top right corner" of the plotted image, and lower values in the "bottom left corner".

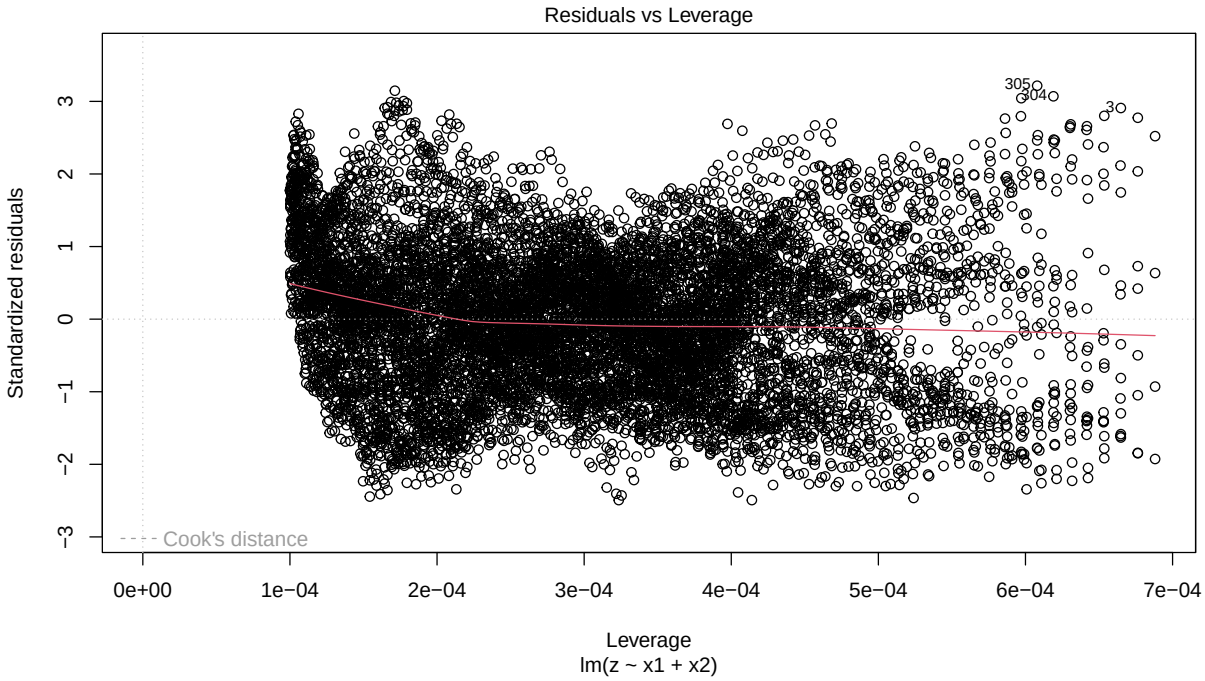
(d)

```
# Assumptions of the linear model
plot(sim1_lm)
```




Q−Q Residuals
Standardized residuals
3872
305
−4
−2
0
2
4
−2
−1
0
1
2
3
Theoretical Quantiles
lm(z ~ x1 + x2)


Scale−Location
√|Standardized residuals|
305
304
3872
−0.5
0.0
0.5
0.0
0.5
1.0
1.5
Fitted values
lm(z ~ x1 + x2)



The residuals vs fitted plot shows a pattern, which indicates that the linear model is not appropriate for this data. The Q-Q plot shows that the residuals are not normally distributed, as they deviate quite a bit. The scale-location plot shows that the residuals are homoscedastic, as there is no clear structure visible. The residuals vs leverage plot shows that there arent any high leverage points.

(e)

We could eprform additive decomposition: $z = \mu + Z(s) + \epsilon$, where μ is the mean, Z(s) is a stationary process, and ϵ is the error term.

```
mu <- mean(sim1_df$z)
```

The mean is -0.1321418.

Problem 2

(a)

```
transect <- read.csv("data/transect.txt", header = TRUE, sep = " ")
skim(transect)
```

Table 1: Data summary

Name	transect
Number of rows	5

Number of columns	2
Column type frequency:	
numeric	2
Group variables	None

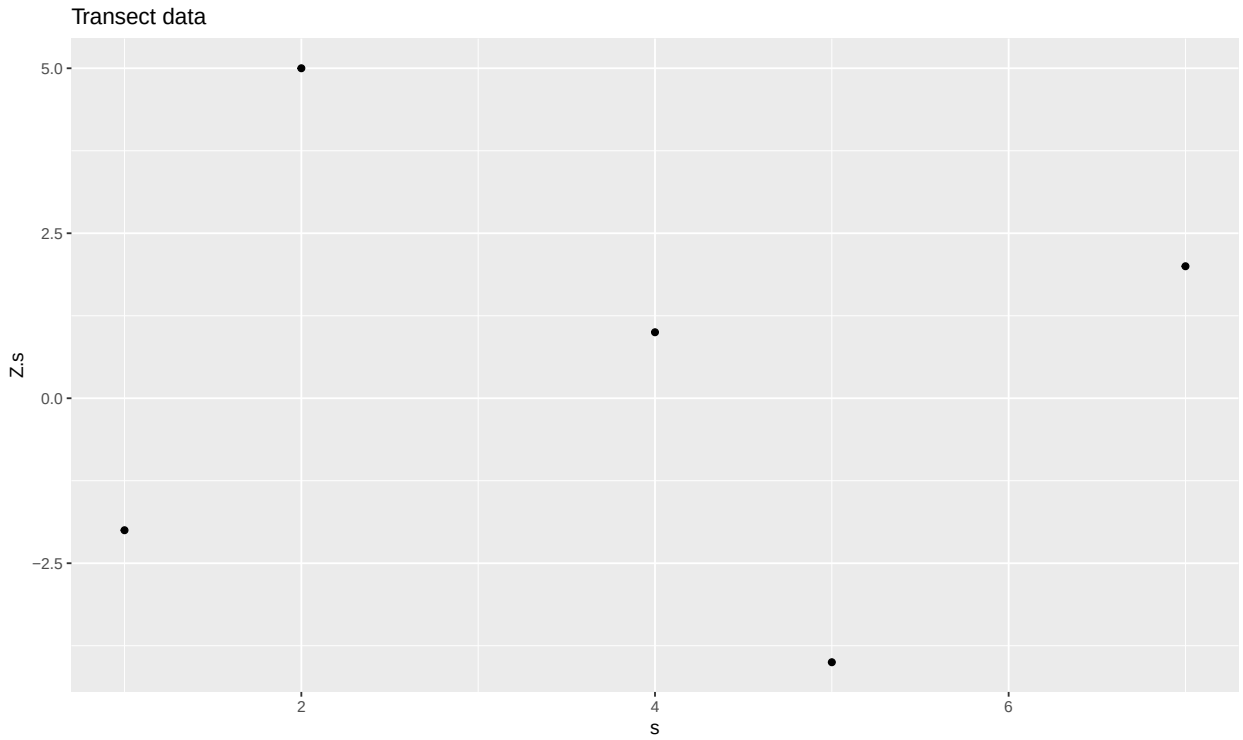
Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
s	0	1	3.8	2.39	1	2	4	5	7	
Z.s	0	1	0.4	3.51	-4	-2	1	2	5	

```
head(transect)
```

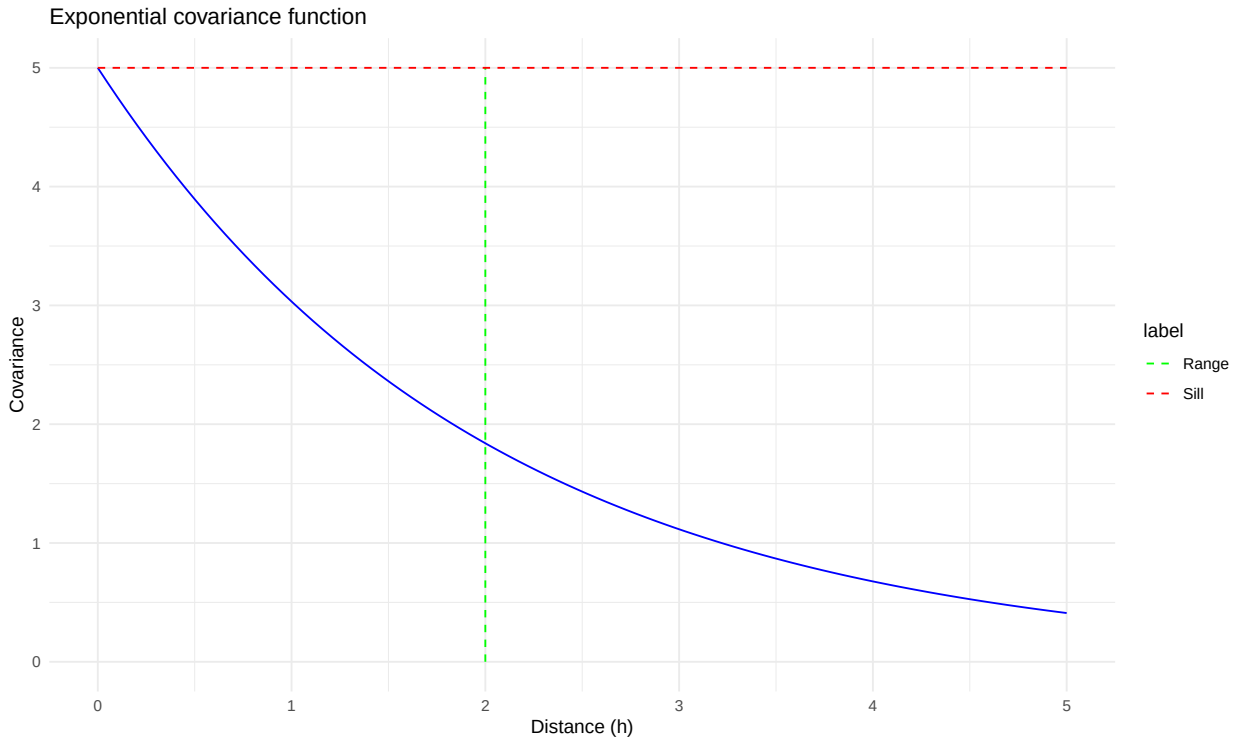
```
##   s Z.s
## 1 1  -2
## 2 2   5
## 3 4   1
## 4 5  -4
## 5 7   2
```

```
ggplot(transect, aes(x = s, y = Z.s)) +
  geom_point() +
  labs(title = "Transect data", x = "s", y = "Z.s")
```



(b)

```r
exp.cov <- function(h, theta2, theta3) {
    return(theta2 * exp(-h / theta3))
}

h <- seq(0, 5, length.out = 100)
theta1 <- 0
theta2 <- 5
theta3 <- 2

# Calculate covariance values
cov_vals <- numeric(length(h))
for (i in 1:length(h)) {
    cov_vals[i] <- exp.cov(h[i], theta2, theta3)
}

sill_line <- data.frame(h = c(0, max(h)), cov = theta2, label = "Sill")
range_line <- data.frame(h = theta3, cov = c(0, theta2), label = "Range")

ggplot(data.frame(h = h, cov = cov_vals), aes(x = h, y = cov)) +
  geom_line(color = "blue") +
  geom_line(data = range_line, aes(x = h, y = cov, color = label), linetype = "dashed") +
  geom_line(data = sill_line, aes(x = h, y = cov, color = label), linetype = "dashed") +
  scale_color_manual(values = c("Sill" = "red", "Range" = "green")) +
  labs(title = "Exponential covariance function",
       x = "Distance (h)",
       y = "Covariance") +
  theme_minimal()
```



Nugget = 0, Partial sill = 5, Range = 2.

(c)

```
dist.matrix <- function(x, y) {
    return(abs(outer(x, y, "-")))
}

DIST.MAT <- dist.matrix(transect$s, transect$s)
DIST.MAT
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    3    4    6
## [2,]    1    0    2    3    5
## [3,]    3    2    0    1    3
## [4,]    4    3    1    0    2
## [5,]    6    5    3    2    0
```

(d)

```
SIGMA <- exp.cov(DIST.MAT, theta2, theta3)
SIGMA
```

```
##           [,1]     [,2]     [,3]      [,4]      [,5]
## [1,] 5.0000000 3.032653 1.115651 0.6766764 0.2489353
```

```
## [2,] 3.0326533 5.000000 1.839397 1.1156508 0.4104250
## [3,] 1.1156508 1.839397 5.000000 3.0326533 1.1156508
## [4,] 0.6766764 1.115651 3.032653 5.0000000 1.8393972
## [5,] 0.2489353 0.410425 1.115651 1.8393972 5.0000000
```

(e)

```
snew <- seq(min(transect$s), max(transect$s), length.out = 10)

dist.new <- dist.matrix(snew, transect$s)
sigma.new <- matrix(0, nrow = length(snew), ncol = length(transect$s))
for (i in 1:length(snew)) {
    for (j in 1:length(transect$s)) {
        sigma.new[i, j] <- exp.cov(dist.new[i, j], theta2, theta3)
    }
}

dist.new
```

```
##            [,1]      [,2]      [,3]      [,4]      [,5]
##  [1,] 0.0000000 1.0000000 3.0000000 4.0000000 6.0000000
##  [2,] 0.6666667 0.3333333 2.3333333 3.3333333 5.3333333
##  [3,] 1.3333333 0.3333333 1.6666667 2.6666667 4.6666667
##  [4,] 2.0000000 1.0000000 1.0000000 2.0000000 4.0000000
##  [5,] 2.6666667 1.6666667 0.3333333 1.3333333 3.3333333
##  [6,] 3.3333333 2.3333333 0.3333333 0.6666667 2.6666667
##  [7,] 4.0000000 3.0000000 1.0000000 0.0000000 2.0000000
##  [8,] 4.6666667 3.6666667 1.6666667 0.6666667 1.3333333
##  [9,] 5.3333333 4.3333333 2.3333333 1.3333333 0.6666667
## [10,] 6.0000000 5.0000000 3.0000000 2.0000000 0.0000000
```

```
sigma.new
```

```
##            [,1]      [,2]     [,3]      [,4]      [,5]
##  [1,] 5.0000000 3.0326533 1.115651 0.6766764 0.2489353
##  [2,] 3.5826566 4.2324086 1.557016 0.9443780 0.3474173
##  [3,] 2.5670856 4.2324086 2.172991 1.3179857 0.4848598
##  [4,] 1.8393972 3.0326533 3.032653 1.8393972 0.6766764
##  [5,] 1.3179857 2.1729910 4.232409 2.5670856 0.9443780
##  [6,] 0.9443780 1.5570161 4.232409 3.5826566 1.3179857
##  [7,] 0.6766764 1.1156508 3.032653 5.0000000 1.8393972
##  [8,] 0.4848598 0.7993987 2.172991 3.5826566 2.5670856
##  [9,] 0.3474173 0.5727942 1.557016 2.5670856 3.5826566
## [10,] 0.2489353 0.4104250 1.115651 1.8393972 5.0000000
```

(f)

```
snew <- seq(min(transect$s), max(transect$s), length.out = 5)
```

```r
DIST.OBS.PRED <- dist.matrix(transect$s, snew)
DIST.OBS.OBS <- dist.matrix(transect$s, transect$s)

COV.OBS.PRED <- exp.cov(DIST.OBS.PRED, theta2, theta3)
COV.OBS.OBS <- exp.cov(DIST.OBS.OBS, theta2, theta3)

COV.OBS.PRED.INV <- solve(COV.OBS.PRED)
COV.OBS.OBS.INV <- solve(COV.OBS.OBS)
```