QSCI482 Lab 3

Simon Hans Edasi

2023-10-13

```
rm(list = ls())
```

Question 1

The number of southern resident killer whales is known to be exactly 75, which allows for accurate testing of different methods of estimating population size. One popular method is line-transect sampling. I'll spare you the details, but 6 surveys were conducted in the range of the killer whales using line-transect sampling, resulting in abundance estimates of 74, 77, 75, 75, 81, and 84. You will be testing whether the sample mean differs from the true population mean of 75.

```
samp <- c(74,77,75,75,81,84)
```

Find the mean, sample standard deviation, and degrees of freedom from the sample.

```
samp_mean <- mean(samp)
samp_std <- sd(samp)
samp_dof <- length(samp) - 1
```

Using the results, calculate the t statistic using the equations in the lectures.

```
mu <- 75
t <- (samp_mean - mu) / (samp_std / sqrt(length(samp)))
print(t)
```

```
## [1] 1.63984
```

Using the t distribution, calculate the probability that the observed t statistic, or a more extreme value, would be observed. If this is smaller than 0.025 you can reject the null hypothesis.

```
pt(t,df = samp_dof, lower.tail = FALSE)
```

```
## [1] 0.08098151
```

Calculate the p-value. Remember that for a one-tailed test the p-value is the area in the tail of interest (for testing the null hypothesis that something is bigger, the p-value is the left tail; if testing the null hypothesis that something is smaller, the p-value is the right tail). For a two-tailed test, it is twice the area of the smallest tail. R has useful functions max() and min() to calculate this.

```
p_left <- pnorm(-t,0,1)      # left rejection region
p <- 2 * p_left                        # p value for a two tail test
print(p)
```

```
## [1] 0.1010384
```

We do not reject the null hypothesis that the sample mean differs from population. Quick sanity check to make sure things are correct

```r
mu <- 75
alpha = 0.05
# define rejection regions
rej_left <- qnorm(alpha/2, 0,1)
rej_right <- qnorm(alpha/2, 0,1, lower.tail = FALSE)
print(rej_left)
```

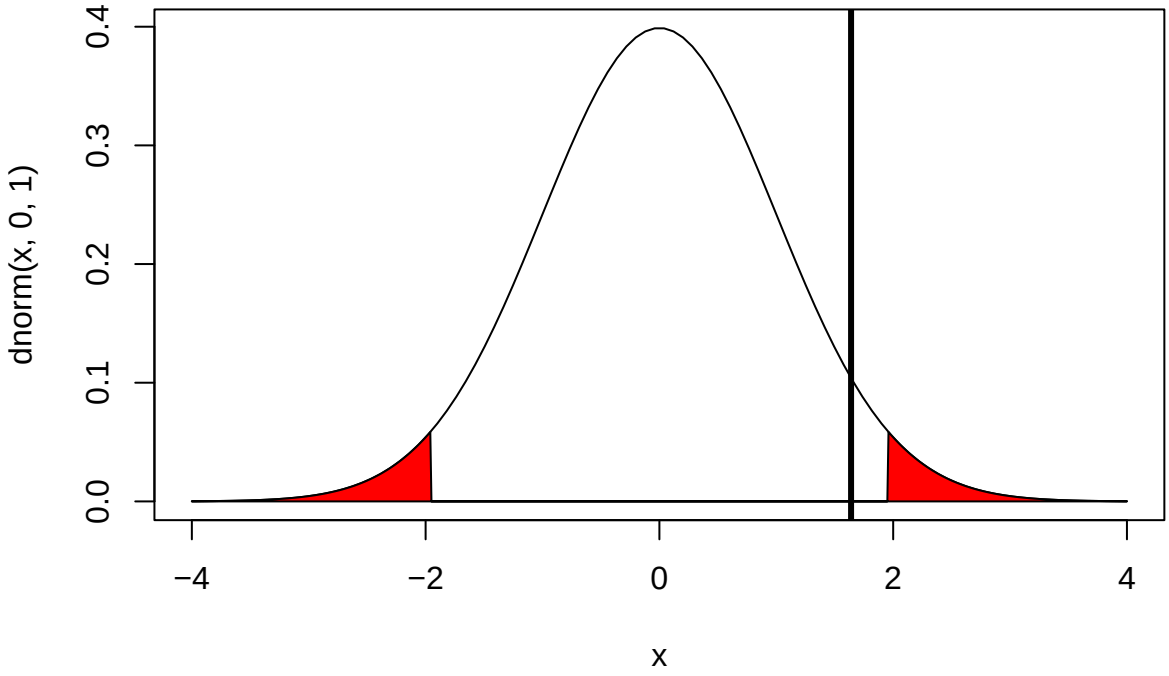
```
## [1] -1.959964
```

```r
#plot the curve with rejection regions
#plot the curve with rejection regions
x <- seq(-4,4,length.out = 100)
plot(
  x,
  y = dnorm(x,0,1),
  type = 'l')
## x and y for the whole area
xReject <- c(seq(-4,4,by=0.01))
yReject <- dnorm(xReject,0,1)

yReject[xReject > rej_left & xReject < rej_right] <- 0

polygon(c(xReject,xReject[length(xReject)],xReject[1]),
        c(yReject,0, 0), col='red')

# plot test statistic
abline(v = t, col = 'black', lwd = 3)
```



Question 2

Google Maps is one of many smartphone apps that people use to get directions. It comes with a built-in predictor of the time it will take to get to your destination. But how accurate are its predictions? Read in the data in GoogleMapTimes.csv Download GoogleMapTimes.csv and save it to an informatively-named variable, e.g. GoogleTimes. Calculate observed minus predicted times and store the vector of answers in a variable (e.g. differences).

```
rm(list = ls())
gt <- read.csv('GoogleMapTimes.csv')
obs <- gt$Actual
pred <- gt$GMapPredicted
diff <- obs - pred
mean(diff)
```

```
## [1] 1.035714
```

Question 3

Assuming that the differences in Question 2 are a random sample from a normal distribution, find the probability of (1) arriving before the predicted time, and (2) arriving more than 5 minutes after the predicted time. For the purposes of this question, assume that the mean and SD of the sample is the same as the population mean μ and population standard deviation σ of the normal distribution.

So we want to calculate the probability of being early (int[-inf, 0]) and the probability of being 5 minutes late (int[5,inf])

```
early <- signif(pnorm(0,mean(diff),sd(diff)), 3)
late <- signif(1-pnorm(5,mean(diff),sd(diff)), 3)
library(glue)
glue('probability of being early is {early} and probability of being 5 minutes late is {late}')
```

```
## probability of being early is 0.347 and probability of being 5 minutes late is 0.0657
```

Question 4

Now, relax the assumption that the sample standard deviation is the same as the population standard deviation. Run a one-sample t-test on the data in Questions 2-3, to *test whether the observed times differ from the predicted times*. Think carefully about what your null hypothesis is, whether it is one-tailed or two-tailed, and how to calculate the p-value. Do you trust Google Maps to give you an accurate answer?

H_0 : Mean travel time difference = 0

H_A : Mean travel time difference $\neq$ 0

Because we are interested if observed is equal to predicted and not greater we will run a two tailed test.

Choose 95%CI – alpha= 0.05

```
# #calculate t and plot it
# obs_mean <- mean(obs)
# obs_std <- sd(obs)
# obs_se <- obs_std / sqrt(length(obs))
#
# pred_mean <- mean(pred)
# pred_std <- sd(pred)
# pred_se <- pred_mean / pred_std
#
#
#
# # pooled_var <- (obs_std^2 + pred_std^2) / 2
# #
# # t <- (pred_mean - obs_mean) / pooled_var
mu <- 0
X <- mean(diff)
samp_std <- sd(diff)
samp_se <- samp_std / sqrt(length(diff))
t <- (X - mu) / samp_se
alpha <- 0.05
# define rejection regions
rej_left <- qnorm(alpha/2, 0,1)
rej_right <- qnorm(alpha/2, 0,1, lower.tail = FALSE)
#plot the curve with rejection regions
x <- seq(-4,4,length.out = 100)
print(t)
```

```
## [1] 2.948929
```

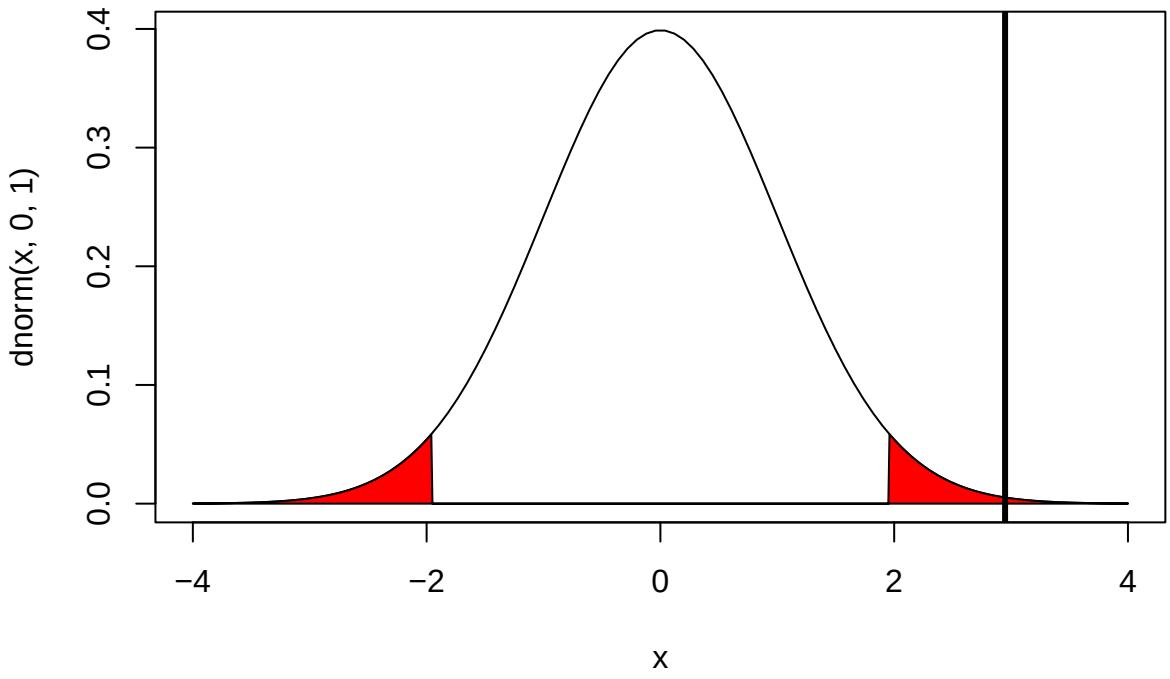
```
plot(
  x,
  y = dnorm(x,0,1),
  type = 'l')
## x and y for the whole area
```

```r
xReject <- c(seq(-4,4,by=0.01))
yReject <- dnorm(xReject,0,1)

yReject[xReject > rej_left & xReject < rej_right] <- 0

polygon(c(xReject,xReject[length(xReject)],xReject[1]),
        c(yReject,0, 0), col='red')

# plot test statistic
abline(v = t, col = 'black', lwd = 3)
```



Determine P-value

```r
p_left <- pt(-t,df = length(obs) - 1)      # left rejection region

p <- 2 * p_left                           # p value for a two tail test
print(p)
```

```
## [1] 0.004674201
```

Our p value is much lower than alpha, so we reject the null that GM travel times equals non GM travel times.Interpreting our t-statistic, we can see that GM predicted travel times are much longer than non GM travel times. Given these results I would not trust google maps