05 Understanding the Effects of Sampling

Contents

(06) Sampling Data

Preamble

```
# Preamble

## Install Pacman
load.pac <- function() {

  if(require("pacman")){
    library(pacman)
  }else{
    install.packages("pacman")
    library(pacman)
  }

    pacman::p_load(xts, sp, gstat, ggplot2, rmarkdown, reshape2, ggmap,
                   parallel, dplyr, plotly, tidyverse, reticulate, UsingR, Rmpfr,
                   swirl, corrplot, gridExtra, mise, latex2exp, tree, rpart, lattice,
                   coin, primes, epitools, maps, clipr, ggmap)


   mise()
   select <- dplyr::select
}

load.pac()
```

```
## Loading required package: pacman
```

```
load(file = "~/Notes/DataSci/ThinkingAboutData/TAD.rdata")
load(file = "./TAD.rdata")
print("Success")
```

```
## [1] "Success"
```

```
knitr::opts_chunk$set(fig.path = "./figure/")
```

Biased

Biased Sampling

Use the crabs data and operate under the assumption that the file contains the entire population of crabs.

Random Sampling

Take a simple random sample of 100 crabs pre-molting size:

```
N <- 100
s <- sample(crabsmolt$presz, size = N, replace = FALSE)
```

Consider the Bias

```
(bias_mean <- mean(crabsmolt$presz)- mean(s))
```

```
## [1] 0.3578644
```

```
(bias_sd <- sd(crabsmolt$presz)- sd(s))
```

```
## [1] 0.8777488
```

Biased Sampling

Say for example that the experiment was such that only crabs $\geq$ 130 mm could be measured, such a sample can be simulated:

```
N <- 100

# Consider the crabs that could have been measured
## Base
a <- crabsmolt$presz[crabsmolt$presz>130]
## Dplyr
a <- crabsmolt %>% dplyr::filter(presz > 130)
a <- a$presz

# Resample the data
s <- sample(a, size = N, replace = FALSE)

# Calculate the bias of the sample mean (i.e. the difference)
(mean_bias <- mean(crabsmolt$presz) - mean(s))
```

```
## [1] -9.786136
```

```r
(sd_bias <- sd(crabsmolt$presz) - sd(s))
```

```
## [1] 10.71388
```

Measuring Type I and Type II Errors

Type I Error

The probability of rejecting the null hypothesis when it is true is the p-value, in this case we will set it to 5%.

So if we drew two samples from the population:

```r
N <- 50

ss <- list()
for (i in 1:2) {
  s <- sample(crabsmolt$presz, size = N, replace = TRUE)
  ss[[i]] <- s
}
```

And then peformed a hypothesis test for the difference in mean values between the samples:

```r
t.test(ss[[1]], ss[[2]])$p.value
```

```
## [1] 0.9344261
```

The p-value is quite high, so the probability of getting a false positive is too high for the null hypothesis to be rejected, however, there will be false positives and this is precisely what the p-value measures, for example, observe that if we repeated this 9999 times, the rate of false positives is equal to the p-value (because the p-value is the probability of a false positive):

```r
vals <- replicate(10^4,
                  {
s1 <- sample(crabsmolt$presz, size = N, replace = TRUE)
s2 <- sample(crabsmolt$presz, size = N, replace = TRUE)
t  <- t.test(s1, s2,  )

t$p.value < 0.05
                  })
mean(vals)
```

```
## [1] 0.0465
```

Or for some other value, say a p-value of 13%:

```r
vals <- replicate(10^4,
                  {
s1 <- sample(crabsmolt$presz, size = N, replace = TRUE)
s2 <- sample(crabsmolt$presz, size = N, replace = TRUE)
t  <- t.test(s1, s2,  )

t$p.value < 0.13
                  })
mean(vals)
```

```
## [1] 0.135
```

Observe that this was a sample were repitition was allowed, If repetition is not allowed the p-value is an over estimation of the probability of a false positive, I'm not totally certain why though.

Type II Error

The two populations of crabs `presz` and `postsz` have different mean values, if they are compared but the null hypothesis (that they're identical) is not rejected then this will by a Type II error (A TrueNeg).

Sampling the data and performing a t-test:

```
# Take Samples from the Population
N <- 50
s1 = sample(crabsmolt$presz, size = N, replace = TRUE)
s2 = sample(crabsmolt$postsz, size = N, replace = TRUE)

# Use a t.test to evaluate the hypothesis
t.test(s1, s2)$p.value < 0.05
```

```
## [1] TRUE
```

If this was repeated many times, the *True Negative Rate* may be determined and will approach the probability of a True Negative (β):

```
vals <- replicate(10 ^ 4,
                  {
                    # Take Samples from the Population
                    beta <- 7/100
                    N <- 50
                    s1 = sample(crabsmolt$presz, size = N, replace = TRUE)
                    s2 = sample(crabsmolt$postsz, size = N, replace = TRUE)

                    # Use a t.test to evaluate the hypothesis
                    t.test(s1, s2)$p.value < 0.05
                  })
1-mean(vals)
```

```
## [1] 0.0065
```

The power is the probability of rejecting the null hypothesis when it is false, in this case the simulation was performed under the assumption that the null hypothesis was false and hence an estimate for the power is the proportion of correctly rejected comparisons made which is:

```
vals %>% mean %>% signif(2)
```

```
## [1] 0.99
```

And so the power of this experiment is 99%.

Paired Data

```
hm <- nzhelmet
head(hm)
```

```
##   Cardboard Metal
## 1       146   145
## 2       151   153
## 3       163   161
```

```
## 4       152   151
## 5       151   145
## 6       151   150
```

```
apply(hm, 2, mean)
```

```
## Cardboard     Metal
##  154.5556  152.9444
```

Measuring P-Value from first Principles

Because Each observation is measured twice (One observation, two features), it isn't possible to randomly permutate the observations in order to measure the probability of a ***False Positiave***. Instead, to address this, randomly select observations to swap between features, this new data set will then represent a random sample under the hypothesis that there is no difference between the two features.

Hence a simulation of what the distribution of differences would look like under the assumption that there is no difference (i.e. assuming null hypothesis is true) would look like the histogram as shown in @ref(fig:hm-null-dist)

```
d <- hm$Cardboard - hm$Metal
n <- nrow(hm)
t0 <- mean(d)/(sd(d)/n^0.5)

vals <- replicate(10^3,
                  {
                     s <- sample(c(-1,1), replace = TRUE, size = n)
                    (t <- mean(s*d)/(sd(s*d)/n^0.5))
                  })

df <- data.frame(PF = vals)
ggplot(df, aes(x = PF)) +
    geom_histogram(aes(y =..density..),
                   colour = "black",
                   fill = "white") +
stat_function(fun = dnorm, args = list(mean = mean(df$PF), sd = sd(df$PF))) +
  theme_bw() +
  labs(x = "Difference", y = "")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
library(gapminder)
library(ggplot2)
library(dplyr)
gapminder %>%
  filter(country == "Australia") %>%
  ggplot(aes(x = year,
             y = lifeExp)) +
  geom_point()
```

Australia's life expectancy has increased a great deal over the past 50 years

(See Figure @ref(fig:gg-oz-plot))

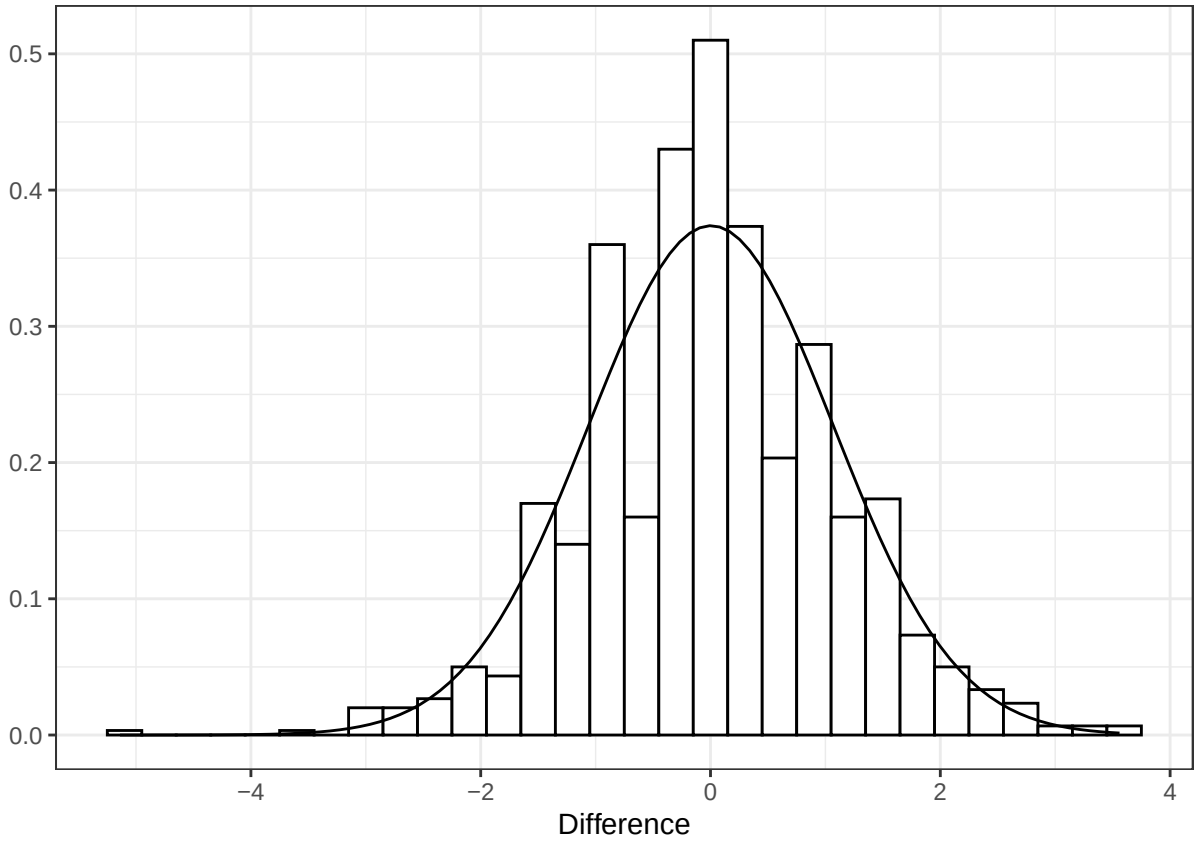

Figure 1: Distribution of Differences Uner the Null Hypothesis

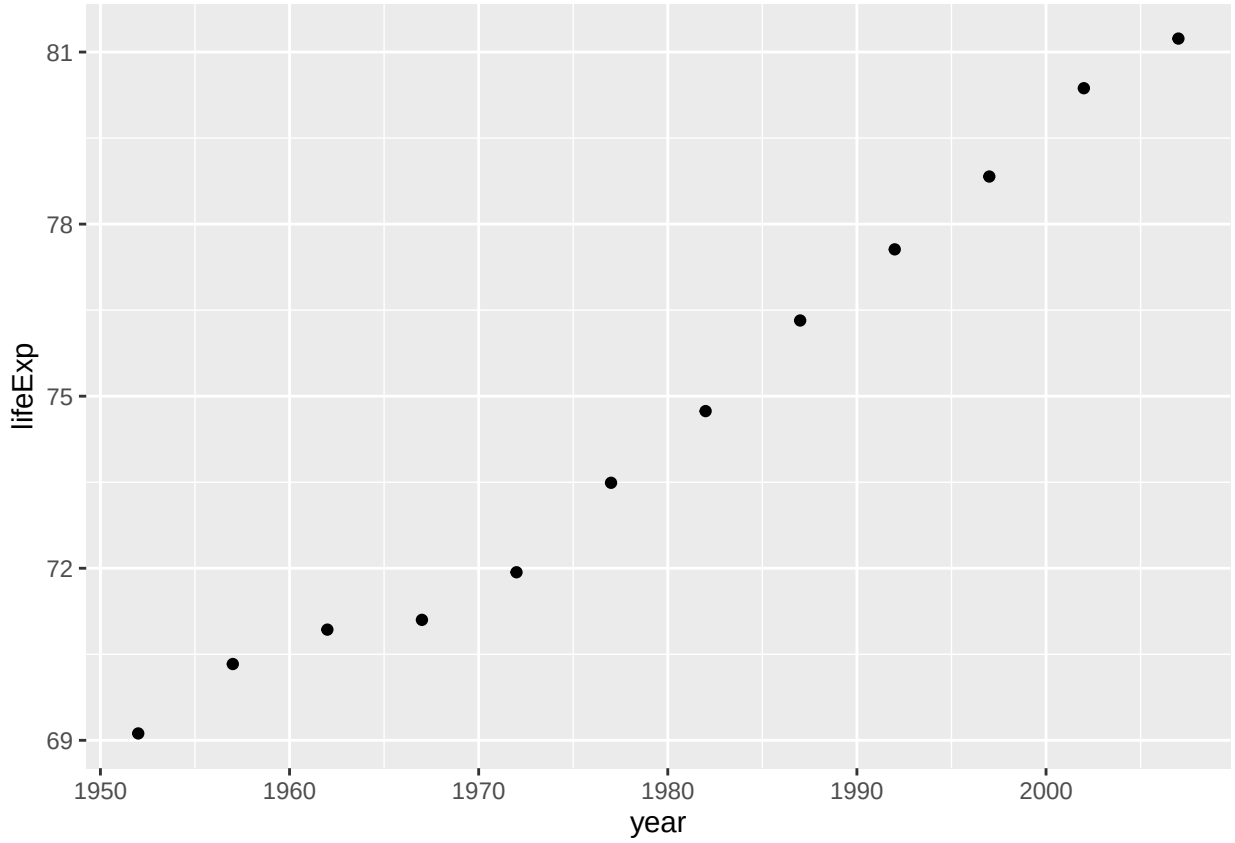

Figure 2: Life expectancy from 1952 - 2007 for Australia. Life expentancy increases steadily except from 1962 to 1969. We can safely say that our life expectancy is higher than it has ever been!