Hefner's Essentials Reference Sheet

Matthew Hefner

November 10, 2018

Contents

1 Sampling

Sampling is collecting a sample of size n of observations from the population. Statistics are a characteristic of a sample; parameters are a characteristic of a population.

2 Notation

Table 1: Notation for Parameters and Statistics

Characteristic	Parameter Notation	Statistic Notation
Proportion	p	$\hat{p}$
Mean	μ	$\hat{x}$
SD	σ	s
Regression Intercept	β_0	$\hat{\beta}_0$ or $\hat{b_0}$
Regression Slope	β_1	$\hat{\beta}_1$ or $\hat{b_1}$

3 Sampling Distributions

- Normal Distribution: Defined by an expected value (mean) and SD).
- Bonomial Distribution: Defined by n number of trials and probability p of success. Mean: np. Var: $np(1-p)$

- Exponential Distribution: Defined by a λ. The PDF is $\lambda e^{-\lambda x}$ for $x \geq 0$. Mean: λ^{-1}. Var: Mean: λ^{-2}

Table 2: Sampling Distribution R Functions

Prefixes (function)	Suffixes (distribution)
Probability Density: d-	Normal: -norm
Cumulative Distribution Function: p-	Binomial -binom
Quantile Function: q-	Exponential: -exp
Random Generation: r-	

4 Probability

Given an event E as a subset of the *sample space* Ω, the three axioms of probability are:

1. $0 \leq P(E) \leq 1$
2. $P(\Omega) = 1$
3. For any sequence of mutually exclusive events $E_1, E_2, \ldots$ (that is, $E_i \cap E_j = \emptyset$ for all $i \neq j$)

$$P(\bigcup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} P(E_i)$$

- The **Conditional Probability** of an event F given E: $P(F|E) = \frac{P(F \cap E)}{P(E)}$.
- **Law of Total Probability and Bayes' Rule** Let $F_1, F_2, \ldots F_n$ be such that $\bigcup_{i=1}^{\infty} F_i = \Omega$ and $F_i \cap F_j = \emptyset$ for all $i \neq j$, with $P(F_i) > 0$ for all i. Then, for any event E,

$$P(E) = \sum_{i=1}^{n} P(E \cap F_i) = \sum_{i=1}^{n} P(E|F_i)P(F_i)$$

$$P(F_i|E) = \frac{P(E \cap F_i)}{P(E)} = \frac{P(E|F_i)P(F_i)}{\sum_{i=1}^{n} P(E|F_i)P(F_i)}$$

5 Regression

The `lm()` function that fits the linear regression model is typically used as `lm(y ~ x, data = data_frame_name)` where:

- y is the outcome variable, followed by a tilde (~).
- x is the explanatory variable(s).

`get_regression_table(model)` from the `moderndive` package can be used on a model from the `lm()` function to get the intercept, associated increases, along with `std_error`, `statistic`, `p_value`, `lower_ci` and `upper_ci` for the model. The *residual* of an observation in a model is $y - \hat{y}$. We want there to be **no systematic pattern to the residuals** (*on average the error is 0* and *the spread of residuals should not depend on x)*. This is important for standard error and confidence intervals to have a meaningful interpretation. A table of observed values (y), values of explanatory variables (x), fitted values ($\hat{y}$), and residuals can be obtained with the `get_regression_points()`, also from the `moderndive` package.

Below is an example of a linear model of one numerical and one categorical explanatory predictor.

```
evals_ch7 <- evals %>%
  select(score, age, gender)
#EDA plot
ggplot(evals_ch7, aes(x = age, y = score, color = gender)) +
  geom_jitter() +
  labs(x = "Age", y = "Teaching Score", color = "Gender") +
  geom_smooth(method = "lm", se = FALSE)
```

```
# Fit Regression Model
score_model_2 <- lm(score ~ age + gender, data = evals_ch7)
# Get Regression Table
regp <- get_regression_table(score_model_2)
# Get Regression Points
get_regression_points(score_model_2)
```

The regression table tells us that the y-intercept of our model is 4.484 for females and 0.191 less than that for males. For both genders, score decreases by -0.009 per year.

This can also be done with an *interaction model* when the effect of one predictor is dependent on that of another. In our example, this means that in addition to seperate intercepts, the slope, change in `score` over change in `age`, is different for each value of `gender`.

```
# Fit Regression Model
score_model_2 <- lm(score ~ age * gender, data = evals_ch7)
# Get Regression Table
get_regression_table(score_model_2)
# Get Regression Points
get_regression_points(score_model_2)
```

6 Infer Pipeline

(1) `specify(data, response ~ explanatory)` or `Specify(data, response = 'response') %>%`

(2) `hypothesize(null = 'null') %>%` where null is ‘point’ point hypotheses involving a single sample or ‘independence’ for testing for independence between two variables

(3) `generate(reps = 'numberOfReps', type = 'type') %>%` where `type` can be ‘permute’, ‘bootstrap’, or ‘simulate’

(4) `calculate(stat = 'stat') %>%` use `?calculate` to find the stat you want to calculate

(5) `visualize()` use `?visualize` to find appropriate method arguments and endpoint arguments for **confidence intervals**.

7 Central Limit Theorem

The means of samples have approximately normal distributions. The larger the sample size, the more normal and the mopre narrow the distribution of the averages becomes and the closer the mean of the samples is to the mean of the population.

8 Bootstrapping

Bootstrapping is a process of sampling with replacement from our original sample to create new *bootstrap samples* of the same size as our original sample. We calculate statistics from these samples and look at their distribution to attempt to determine, with some level of confidence, an interval on which a parameter for the population exists. As an example, the means of of `age_in_2011` of 1000 samples with replacement from `pennies_sample`.

```
bootstrap_distribution <- pennies_sample %>%
  specify(response = age_in_2011) %>%
  generate(reps = 1000) %>%
  calculate(stat = "mean")
```

9 Confidence Intervals

The distribution of the calculated sample statistics can tell us with some level of confidence where the corresponding parameter for the population exists. This is the confidence interval for some % confidence called the *confidence level*.

The mean parameter of the population of pennies can be determined with some % confidence using the `bootstrap_distribution` from above as follows:

```
#First argument is the calculated statistic distribution
#Second argument is the level of confidence
ci <- conf_int(bootstrap_distribution, level = 0.90)
ci
```

```
## # A tibble: 1 x 2
##     `5%` `95%`
##   <dbl> <dbl>
## 1  21.4  28.4
```

Thus, we can say that we are 90% confident that the true mean age of the population (pennies in circulation) is between 21.35 and 28.4025.

10 Hypothesis Testing

> The **p-value** is the probability of observing something more or as extreme as our actual observed value given that the null hypothesis is true.

We assume that the null hypothesis is true and simulate several random permutations of the data. From here, we can calculate sample statistics and then determine our (empirical) p-value.

```
observed #Calculate observed value; we'll assume it's in this variable for brevity
N <- 10^4 - 1 # number of times to repeat the process
result <- numeric(N) # space to save the random differences
for (i in 1:N){ # sample of size 5, from 1 to 10, without replacement
  index <- sample(10, size = 5, replace = FALSE)
  result[i] <- mean(Worms2[index]) - mean(Worms2[-index])
}
pvalue <- (sum(result >= observed) + 1)/(N + 1) # empirical p-value
pvalue # results would vary since we're randomly sampling
```

If the p-value is less than 0.05 (<5% probability) (unless told a different signifigance level α), we reject the null hypothesis.

> **Type 1 error** occurs upon rejection of a null hypothesis that is actually true
>
> **Type 2 error** occurs when we do not reject a null hypothesis that is false.

11 Goodness of Fit

Chi-squared tests test the fit of a model of probabilities over some number of a categories k given some observation.

```r
#Observation
Observed <- c(30, 14, 34, 45, 57, 20)
names(Observed) <- c("Monday", "Tuesday", "Wednesday", "Thursday",
    "Friday", "Saturday")
#Probability Model to be tested
p = c(1/10, 1/10, 0.15, 1/5, .30, .15)
#Expected is sum(Observed) * p
#Test of fit; null hypothesis: it fits.  Use Alpha to reject or fail to reject.
chisq.test(Observed, p = c(1/10, 1/10, 0.15, 1/5, .30, .15))
```

```
##
##  Chi-squared test for given probabilities
##
## data:  Observed
## X-squared = 11.442, df = 5, p-value = 0.04329
```