Master Guide - Chapter 1: Foundations of Statistical Inference

DANA-4810 - Compiled Guide

2025-10-10

Contents

```
flowchart TD
  A[Start: What are you estimating or testing?] --> B{Parameter of interest}
  B -->|Mean of one population| C{Is sigma known?}
  C -->|Yes| D[Use z CI or z-test]
  C -->|No| E[Use t CI or t-test]
  B -->|Difference of two means| F{Samples paired or independent?}
  F -->|Paired| G[Use paired t-test / paired CI]
  F -->|Independent| H{Are variances equal?}
  H -->|Yes| I[Pooled t-test / pooled CI]
  H -->|No| J[Welch's t-test / unequal-variance CI]
  B -->|Proportion| K[Use proportion CI/test (not covered here)]
  D --> L[Report CI/test-stat & p-value, interpret in context]
  E --> L
```

Notes: The Mermaid flowchart above provides the high-level decision tree for Chapter 1 topics covered in the materials.

Topics covered

- Normal distribution: probabilities and quantiles
- Confidence intervals: z-based and t-based for means
- Hypothesis testing: one-sample, two-sample (paired and independent)
- Variance comparison (F-test)
- Practical workflows and R code used in the course materials

Decision Tree - choosing the right test

This compact decision tree walks through the questions to ask when you have data and want to decide which inference method to use.

```
flowchart TD
  Start([Start]) --> Q1{Parameter of interest?}
  Q1 -->|Mean (one sample)| A1[Is population sigma known?]
  A1 -->|Yes| ZCI[Use z CI / z-test]
  A1 -->|No| TCI[Use t CI / one-sample t-test]
  Q1 -->|Difference of means| Q2{Samples paired?}
  Q2 -->|Yes| Paired[Use paired t-test / CI]
  Q2 -->|No| VarCheck{Are variances equal?}
  VarCheck -->|Yes| Pooled[Use pooled t-test (var.equal=TRUE)]
  VarCheck -->|No| Welch[Use Welch's t-test (default)]
  Q1 -->|Compare variances| Ftest[Use var.test (F-test)]
  Q1 -->|Proportion| Prop[Use proportion CI/test]
  ZCI --> End([Report CI, test-statistic, p-value, interpret])
  TCI --> End
  Paired --> End
  Pooled --> End
  Welch --> End
  Ftest --> End
  Prop --> End
```

Section 1: Normal Distribution (Probability calculations)

Purpose: compute probabilities, quantiles, and values for normally-distributed variables.

Key functions: - pnorm(q, mean = 0, sd = 1) - cumulative probability P(X <= q). - qnorm(p, mean = 0, sd = 1) - quantile: value x with P(X <= x) = p. - dnorm(x, mean = 0, sd = 1) - density (rarely used directly in examples).

When to use: when X is (approximately) Normal with known or assumed mean and sd.

Example: X ~ N(80, 5). Compute P(X <= 60).

```
mu <- 80
sigma <- 5
p_le_60 <- pnorm(60, mean = mu, sd = sigma)
p_le_60
```

```
## [1] 3.167124e-05
```

Explanation of inputs/outputs: - Inputs: numeric ‘q’ (here 60), ‘mean’ (mu), ‘sd’ (sigma). Must be finite numbers; ‘sd’ > 0. - Output: probability in [0,1]. Interpretation: proportion of population ≤ 60.

Common variants: - P(a <= X <= b) = pnorm(b, mu, sigma) - pnorm(a, mu, sigma) - Finding cutoff x for top 1%: qnorm(0.99, mu, sigma)

Section 2: Confidence Intervals for a Population Mean

When: estimating a population mean. Decision split:

Contract:

z-based CI (sigma known): Formula (display): Formula (inline math): $\bar{x} \pm z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}$

R helper (cleaned):

```
norm_interval <- function(xbar, sigma, n, conf.level = 0.95){
  if(!is.numeric(xbar) || !is.numeric(sigma) || !is.numeric(n)) stop("xbar, sigma, and n must be numeri
  alpha <- 1 - conf.level
  z <- qnorm(1 - alpha/2)
  se <- sigma / sqrt(n)
  c(lower = xbar - z * se, upper = xbar + z * se)
}
# example
norm_interval(75, sigma = 5, n = 10, conf.level = 0.90)
```

```
##    lower    upper
## 72.39926 77.60074
```

t-based CI (sigma unknown): use t.test on raw data or compute using qt.

```
t_interval_from_data <- function(x, conf.level = 0.95){
  n <- length(x)
  xbar <- mean(x)
  s <- sd(x)
  alpha <- 1 - conf.level
  t_q <- qt(1 - alpha/2, df = n - 1)
  se <- s / sqrt(n)
  c(lower = xbar - t_q * se, upper = xbar + t_q * se)
}

  # Example using the course ATTENTIMES snippet
  ATTENTIMES <- read.csv("ATTENTIMES.csv", header=TRUE)
  attach(ATTENTIMES)
  # View(ATTENTIMES)
  head(ATTENTIMES)
```

```
##   Attention.Time
## 1           20.7
## 2           23.5
## 3           10.9
## 4           44.1
## 5           15.7
## 6           14.0
```

```
summary(ATTENTIMES)
```

```
##  Attention.Time
##  Min.   : 0.80
##  1st Qu.:10.68
##  Median :19.65
##  Mean   :20.85
##  3rd Qu.:29.02
##  Max.   :48.20
```

```
# Large-Sample Normal (z)Statistic as in the course Rmd
norm.interval = function(Attention.Time, variance = var(Attention.Time), conf.level = 0.99) {
  z = qnorm((1 - conf.level)/2, lower.tail = FALSE)
  xbar = mean(Attention.Time)
  sdx = sqrt(variance/length(Attention.Time))
  c(xbar - z * sdx, xbar + z * sdx)
}

norm.interval(Attention.Time)
```

```
## [1] 15.96165 25.73435
```

```
# Or using the formula shown in the course Rmd
xbar = mean(Attention.Time)
seA = sd(Attention.Time) / sqrt(length(Attention.Time))
z = qnorm(.995)
CIAttention = xbar + c(-1, 1) * z * seA
print(CIAttention)
```

```
## [1] 15.96165 25.73435
```

```
detach(ATTENTIMES)
```

Inputs/Outputs explanation: - For t_interval_from_data: input ‘x’ is a numeric vector; output: named vector with ‘lower’ and ‘upper’. - Edge cases: $n < 2 \rightarrow$ sd undefined, function should error.

Alternative: use built-in t.test(x, conf.level = 0.95) which returns detailed object with statistic, df, CI, and estimate.

```
# Example with t.test
# t.test(att$Attention.Time, conf.level = 0.99)   # if column named Attention.Time
```

Section 3: Hypothesis Testing for a Mean (one-sample)

Decisions: - Use z-test if sigma known (rare). - Use t-test if sigma unknown.

Typical call (raw data): - t.test(x, mu = 0, alternative = c(“two.sided”,“less”,“greater”))

Example using BONES dataset (column LWRATIO):

```
BONES <-read.csv("BONES.csv", header=TRUE)
attach(BONES)
# View(BONES)
head(BONES)
```

```
##   Ratio
## 1 10.73
## 2  8.89
## 3  9.07
## 4  9.20
## 5 10.33
## 6  9.98
```

```
summary(BONES)
```

```
##      Ratio
##  Min.   : 6.230
##  1st Qu.: 8.710
##  Median : 9.200
##  Mean   : 9.258
##  3rd Qu.: 9.930
##  Max.   :12.000
```

```
t.test(Ratio, mu = 8.5)
```

```
##
##  One Sample t-test
##
## data:  Ratio
## t = 4.0303, df = 40, p-value = 0.0002427
## alternative hypothesis: true mean is not equal to 8.5
## 95 percent confidence interval:
##  8.877669 9.637453
## sample estimates:
## mean of x
##  9.257561
```

```
detach(BONES)
```

Outputs and interpretation: - t_res$statistic : thet - value. - t_res$parameter: degrees of freedom (n-1). - t_res$p.value : p-value.-t_res$conf.int: CI for mean. - Decision rule: compare p-value to alpha or use critical t.

Section 4: Two-sample inference (independent and paired)

Cases: 1) Independent samples: check whether variances equal → choose pooled t-test (var.equal = TRUE) or Welch's t-test (default var.equal = FALSE). 2) Paired samples: use paired = TRUE in t.test.

Typical hypotheses (independent samples):

$H_0 : \mu_1 - \mu_2 = 0$ $H_a : \mu_1 - \mu_2 \neq 0$ (or $<$ or $>$)

For paired samples (paired differences d):

$H_0 : \mu_d = 0$ $H_a : \mu_d \neq 0$ (or $>$ or $<$)

Common functions: - t.test(y ∼ group, data = df, var.equal = TRUE/FALSE) - t.test(x, y, paired = TRUE/FALSE)

Example: DIETS

```
DIETS <-read.csv("DIETS.csv", header=TRUE)
attach(DIETS)
# View(DIETS)
head(DIETS)
```

```
##     DIET WTLOSS
## 1 LOWFAT      8
## 2 LOWFAT     10
## 3 LOWFAT     10
## 4 LOWFAT     12
## 5 LOWFAT      9
## 6 LOWFAT      3
```

```
mean(WTLOSS[DIET=="LOWFAT"])
```

```
## [1] 9.31
```

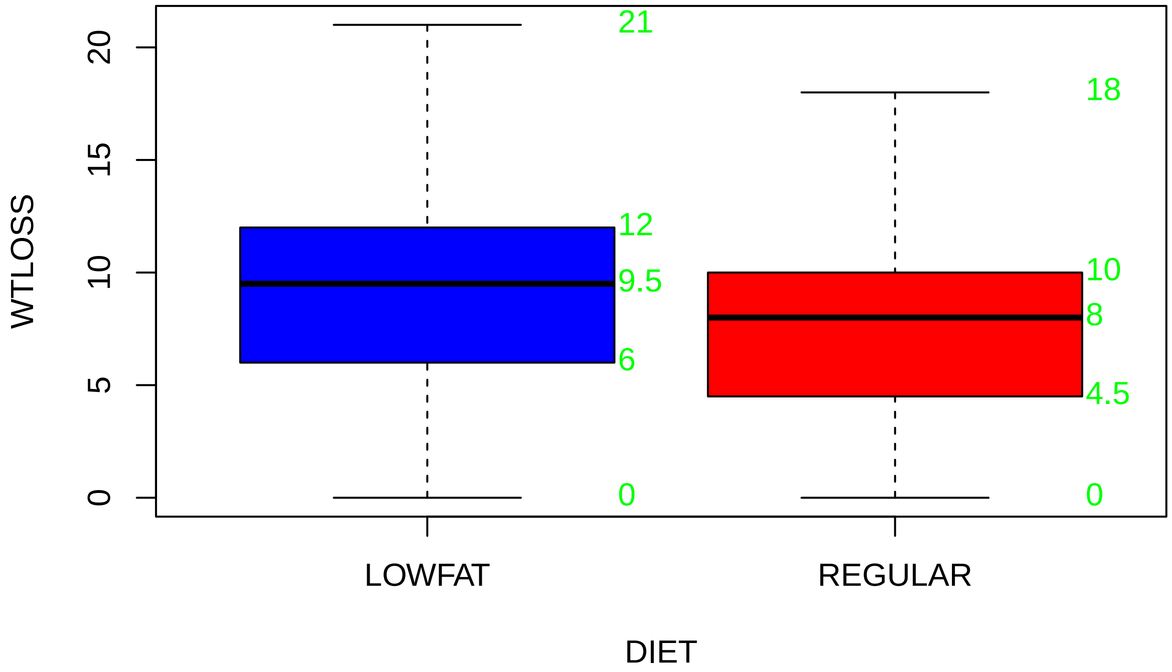
```
mean(WTLOSS[DIET=="REGULAR"])
```

```
## [1] 7.4
```

```
summary(DIETS)
```

```
##      DIET               WTLOSS
##  Length:200         Min.   : 0.000
##  Class :character   1st Qu.: 5.000
##  Mode  :character   Median : 8.500
##                     Mean   : 8.355
##                     3rd Qu.:11.000
##                     Max.   :21.000
```

```
# side-by-side box plots
boxplot(WTLOSS~DIET, col = c("blue","red"))
A=tapply(WTLOSS, DIET,fivenum)
for(i in 1:length(A)){
        text(x=i+0.35,y= A[[i]],labels= A[[i]], pos = 4, offset = 0.7, col = "green")}
```



Paired example: PAIRED.csv

```
PAIRED<- read.csv("PAIRED.csv", header=TRUE)
attach(PAIRED)
head(PAIRED)
```

```
##   PAIR NEW STANDARD
## 1    1  77       72
## 2    2  74       68
## 3    3  82       76
## 4    4  73       68
## 5    5  87       84
## 6    6  69       68
```

```
test=t.test(NEW, STANDARD, conf.level=0.95, alternative = c("greater"), paired=TRUE)
test
```

```
##
##  Paired t-test
##
## data:  NEW and STANDARD
## t = 7.3438, df = 7, p-value = 7.838e-05
## alternative hypothesis: true mean difference is greater than 0
## 95 percent confidence interval:
##  3.246316      Inf
## sample estimates:
## mean difference
##           4.375
```

```
detach(PAIRED)
```

Inputs/Outputs details: - Input: two numeric vectors (or formula with grouping variable). Groups should be coded clearly. - Output: t-test object; check p-value and CI.

Edge cases: - Unequal sample sizes and non-normal data → rely on large-sample approximations or non-parametric tests (not covered fully here).

Section 5: Variance comparison and F-test

Purpose: test $H_0 : \sigma_1^2 = \sigma_2^2$ (compare two population variances). Function: var.test(x, y, alternative = "two.sided")

Typical test statistic (F): $F = \frac{s_1^2}{s_2^2}$ where s_1^2 and s_2^2 are sample variances.

Interpretation: returns F statistic, df for numerator and denominator, p-value.

Example (READING groups):

```
READING <-read.csv("READING.csv", header=TRUE)
attach(READING)
head(READING)
```

```
##   METHOD SCORE
## 1    NEW    70
## 2    NEW    85
## 3    NEW    80
## 4    NEW    76
## 5    NEW    80
## 6    NEW    66
```

```
# Example variance test as shown in the course Rmds
var.test(SCORE[METHOD=="STD"], SCORE[METHOD=="NEW"], alternative = "two.sided")
```

```
##
##  F test to compare two variances
##
## data:  SCORE[METHOD == "STD"] and SCORE[METHOD == "NEW"]
## F = 1.1821, num df = 11, denom df = 9, p-value = 0.8148
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.3021557 4.2410955
## sample estimates:
## ratio of variances
##           1.182056
```

```r
detach(READING)
```

Section 6: Helpful Utilities and Best Practices

- Always inspect data with head(), summary(), boxplot(), and q-q plots (qqnorm + qqline) before testing.
- Use mosaic::favstats() or tapply() to get group summaries.
- For reproducibility, avoid attach(); prefer reading and using data frames directly.

Example workflow for a two-sample comparison: 1. Read data: df <- read.csv("file.csv") 2. Inspect: summary(df); boxplot(y ~ group, data = df) 3. Check normality: qqnorm(…); qqline(…) 4. Check variances: var.test(…) 5. Choose test: pooled or Welch or paired 6. Run t.test and interpret t, df, p-value, CI

Appendix: Cleaned, annotated versions of snippets from course files

z test (summary data) - using BSDA::z.test / zsum.test

- z.test(x, mu, sigma.x, conf.level) expects a numeric vector x and known sigma.x.
- zsum.test(mean.x, n.x, sigma.x, mu) works with summarized statistics.

```r
# Example usage (not run unless package installed):
# library(BSDA)
# z.test(x = some_vector, mu = 0, sigma.x = known_sigma, conf.level = 0.95)
```

Try it (quick checks)

Run the following to see the core calculations from the course files cleaned up (optional):

```r
# Normal probability example
pnorm(60, 80, 5)
```

```
## [1] 3.167124e-05
```

```r
# z vs t quantiles
qnorm(.995)
```

```
## [1] 2.575829
```

```r
qt(.995, df = 4)
```

```
## [1] 4.604095
```

```r
# small t-interval demo
set.seed(1)
x <- rnorm(5, mean = 75, sd = 5)
t.test(x)
```

```
##
##  One Sample t-test
##
## data:  x
## t = 35.202, df = 4, p-value = 3.887e-06
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  69.67991 81.61279
## sample estimates:
## mean of x
##  75.64635
```

Completion notes

- I created this single consolidated R Markdown file for Chapter 1. It uses Mermaid for flowcharts which will render in HTML output (and some RStudio viewers). If you want PNG/SVG flowcharts embedded for PDF knitting, I can add a pre-rendering step.
- Next steps I can take: generate separate `Master_Guide_Chapter1.pdf` or add more worked examples from each dataset in the folder. Tell me if you want the latter.