Summarization and Bivariate Analysis
Week 4

PH 700A, Spring 2025

Rick Calvo

Table of contents

0.1 Recap

Last week: Data Management and Manipulation

Today: Basic Descriptive Analysis

0.2 Understanding Your Data

- You must understand your data before you can perform an analysis
- Often, the data arrangement determines the analysis plan
- Poor preparation leads to wasted time

0.3 Simple Data Analysis

DATA ANALYSIS
CRUYFF'S SIMPLE ~~FOOTBALL~~
"PLAYING ~~FOOTBALL~~ DATA ANALYSIS IS VERY SIMPLE, BUT PLAYING SIMPLE ~~FOOTBALL~~ DATA ANALYSIS IS THE HARDEST THING THERE IS."

0.4 Packages

- `gmodels`
- `stats`
- `gtsummary`
- `e1071`
- `ggplot2`
- `car`

0.5 Special Function for `NA`

To identify missing values, R typically does not interpret `variable == NA` properly.

In order to identify `NA` values, the function is `is.na(variable`.

To identify *present* values (i.e. non-NA values), you have to use the *NOT* operator `!`.

```
!is.na(variable)
```

0.6 MIMIC Data Usage

```
df.dxwide <- df.dx %>%
  pivot_wider(
    id_cols = c(subject_id, stay_id),
    names_from = seq_num,
    values_from = c(icd_code, icd_version, icd_title)
  )

df.ed <- df.ed %>% mutate(hlos = outtime-intime)

df.ed <- df.ed %>% mutate(across(c("gender", "race"), as.factor))

df.triage <- df.triage %>% mutate(fever = if_else(temperature >= 100.4, 1, 0))
```

0.7 Categorical Data

- Usually represented as frequencies and percentages
- Chi-square tests used to evaluate two categorical variables
- Fisher's exact tests for small sample sizes
- Any number of categories can be evaluated

0.8 Basic Code

0.8.1 Frequencies Crosstab and Chi-square Test

```
library(stats)
library(gmodels)

crosstab1 <- table(df$var1, df$var2, useNA = "ifany")

CrossTable(crosstab1)
```

```r
chisq.test(df$var1, df$var2, correct=FALSE)
```

- `var1` and `var2` should be categorical variables. `var1` will be the *rows* and `var2` will be the *columns* of the table
- `useNA = "ifany"` will add missing values to the frequency table if they exist. Other options include `"no"` and `"always"`
- `correct = FALSE` applies a continuity correction for use when a categorical variable was derived from a continuous variable

0.9 Example w/ MIMIC Data

```r
# Dichotomizing Race to White/Non-White
df.ed <- df.ed %>%
  mutate(race_white = case_when(
    race == "WHITE" ~ 1,
    race == "WHITE - BRAZILIAN" ~ 1,
    race == "WHITE - OTHER EUROPEAN" ~ 1,
    .default = 0
))

library(stats)
library(gmodels)

table(df.ed$race_white, df.ed$gender, useNA = "ifany")
```

```
     F  M
  0 64 16
  1 58 84
```

```r
chisq.test(df.ed$race_white, df.ed$gender, correct=FALSE)
```

```

	Pearson's Chi-squared test

data:  df.ed$race_white and df.ed$gender
X-squared = 31.692, df = 1, p-value = 1.807e-08
```

Note: A warning will be given when cell sizes are small but the analysis will proceed nonetheless.

0.10 CrossTable Command from `library(gmodels)`

```r
library(gmodels)

CrossTable(x, y, ...)
```

`x` refers to your row object

`y` refers to your column object

By default, `CrossTable` will only output frequencies and percentages.

To perform a statistical analysis, you must specify:

- `chisq = TRUE` for standard chi-square test
- `fisher = TRUE` for Fisher's exact test – for low cell sizes
- `mcnemar = TRUE` for paired data

0.11 CrossTable Example

```r
library(gmodels)

CrossTable(df.ed$race_white, df.ed$gender, chisq = TRUE)
```

```
   Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|
Total Observations in Table:  222


                 | df.ed$gender 
df.ed$race_white |         F |         M | Row Total | 
-----------------|-----------|-----------|-----------|
               0 |        64 |        16 |        80 | 
                 |     9.131 |    11.140 |           | 
                 |     0.800 |     0.200 |     0.360 | 
                 |     0.525 |     0.160 |           | 
                 |     0.288 |     0.072 |           | 
-----------------|-----------|-----------|-----------|
               1 |        58 |        84 |       142 | 
                 |     5.144 |     6.276 |           | 
                 |     0.408 |     0.592 |     0.640 | 
                 |     0.475 |     0.840 |           | 
                 |     0.261 |     0.378 |           | 
-----------------|-----------|-----------|-----------|
    Column Total |       122 |       100 |       222 | 
                 |     0.550 |     0.450 |           | 
-----------------|-----------|-----------|-----------|


Statistics for All Table Factors


Pearson's Chi-squared test 
------------------------------------------------------------
Chi^2 =  31.69161     d.f. =  1     p =  1.807007e-08 

Pearson's Chi-squared test with Yates' continuity correction 
------------------------------------------------------------
Chi^2 =  30.12962     d.f. =  1     p =  4.041116e-08 
```

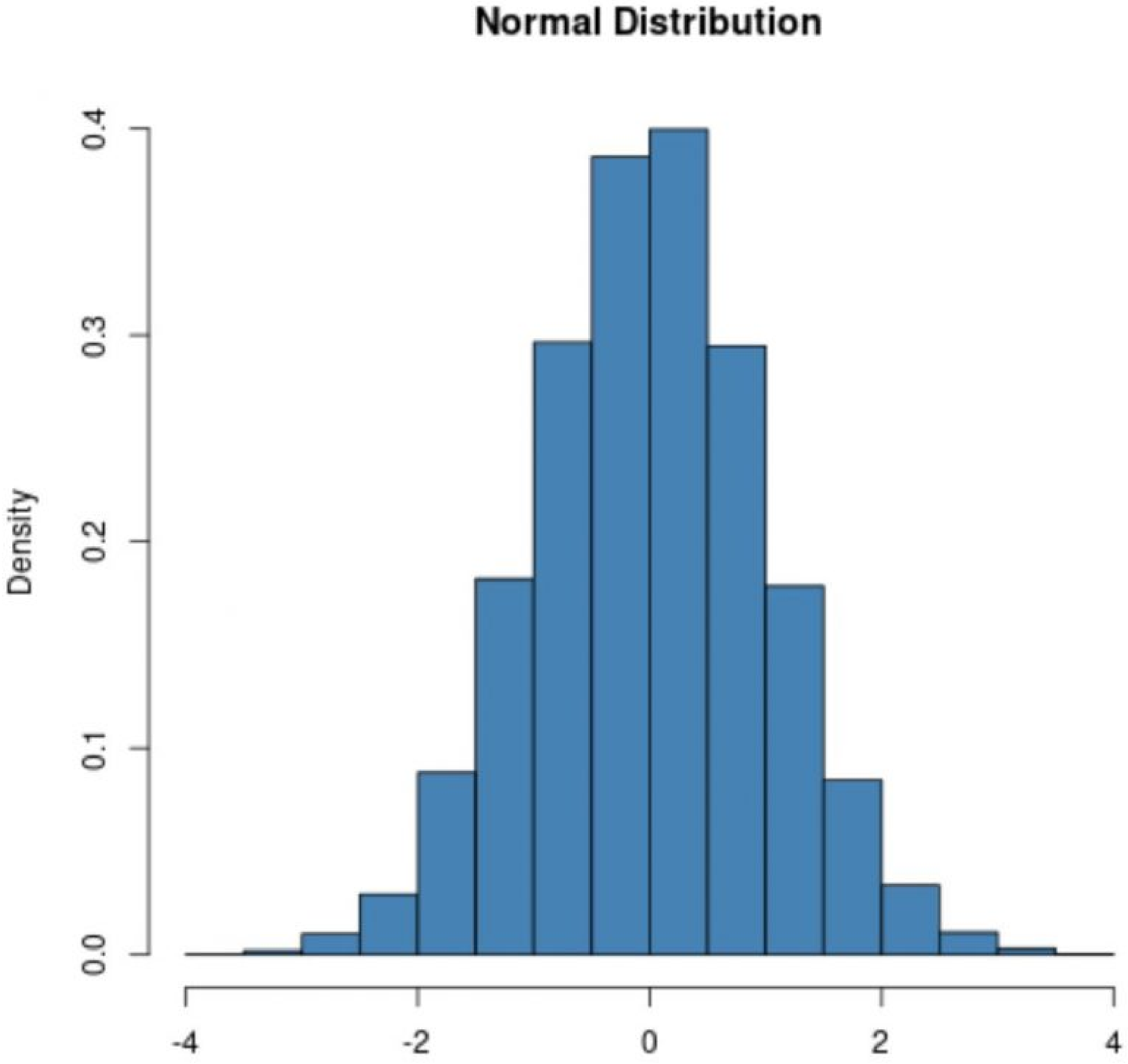
0.12 Bivariate Data Analysis With Continuous Variables

Dependent Variable = Continuous

- Independent Variable = Continuous
- Pearson correlation

- Spearman correlation
- Independent Variable = Categorical
- t-test
- ANOVA
- Mann-Whitney U (Rank Sum)
- Kruskal-Wallis Test

0.13 Behold, the Distribution!



Distributional assumptions of normality must be evaluated!

1 Basic Analysis

1.1 Single Variable Univariate Statistics

Using `tidyverse` and the `summarise()` command, we can output summary statistics of continuous variables

```
df.ed %>%
    summarise(
      obs    = n(),
      min  = min(hlos),
      max  = max(hlos),
      mean = mean(hlos),
      sd   = sd(hlos),
      median  = median(hlos),
      p25  = quantile(hlos, probs = 0.25),
      p75  = quantile(hlos, probs = 0.75)
    )
```

```
  obs    min       max          mean       sd      median          p25
1 222 4 mins 4460 mins 485.7803 mins 509.7573 350.5 mins 253.25 mins
            p75
1 526.0875 mins
```

1.2 Tidyverse Scoped Verbs

Tidyverse can incorporate “scoped verbs” to operate on multiple variables in one block of code. These include `across()`, `pick()`, and a modified `summarise_all()` to apply a function across a selection/all variables in a data frame.

```
df.ed %>%
  select(hlos) %>%
    summarise_all(list(
      n = length,
      min = min,
      max = max,
      mean = mean,
      sd = sd,
      median = median,
      p25 = ~quantile(., probs = 0.25),
      p75 = ~quantile(., probs = 0.75)
    ))
    n    min       max         mean        sd      median         p25
1 222 4 mins 4460 mins 485.7803 mins 509.7573 350.5 mins 253.25 mins
            p75
1 526.0875 mins
```

Note that the `quantile` call is a function that can be used for any value, so it requires a `~` operator to tag it as a function. This comes from the `purrr` package within `tidyverse` that contains additional programming shorthand tools.

1.3 Many univariates at once

```
df.ed %>%
  select(hlos, intime, outtime) %>%
    summarise_all(list(
      n = length,
      min = min,
      max = max,
      mean = mean,
      sd = sd,
      median = median,
      p25 = ~quantile(., probs = 0.25),
      p75 = ~quantile(., probs = 0.75)
    ))
```

```
  hlos_n intime_n outtime_n hlos_min          intime_min         outtime_min
1    222      222       222   4 mins 2112-09-17 18:46:00 2112-09-17 19:50:00
   hlos_max          intime_max         outtime_max     hlos_mean
1 4460 mins 2201-10-30 10:48:00 2201-10-30 12:25:00 485.7803 mins
          intime_mean        outtime_mean  hlos_sd intime_sd outtime_sd
1 2157-09-10 17:55:54 2157-09-11 02:01:41 509.7573 700146546  700150031
  hlos_median       intime_median      outtime_median    hlos_p25
1  350.5 mins 2150-03-08 17:20:00 2150-03-08 23:59:54 253.25 mins
           intime_p25         outtime_p25      hlos_p75          intime_p75
1 2142-05-15 16:41:45 2142-05-15 23:50:30 526.0875 mins 2177-11-24 14:14:00
          outtime_p75
1 2177-11-24 19:51:10
```

Note: `summarise` and `summarise_all` will not work if missing values are present. A straight `tidyverse` solution requires advanced programming.

Characteristic	N = 222[1]
temperature	98.10 (97.60, 98.50)
Unknown	26
heartrate	90 (77, 104)
Unknown	24
resprate	18 (16, 18)
Unknown	23
o2sat	98 (97, 100)
Unknown	24
sbp	136 (116, 155)
Unknown	23
dbp	72 (62, 83)
Unknown	23

[1] Median (Q1, Q3)

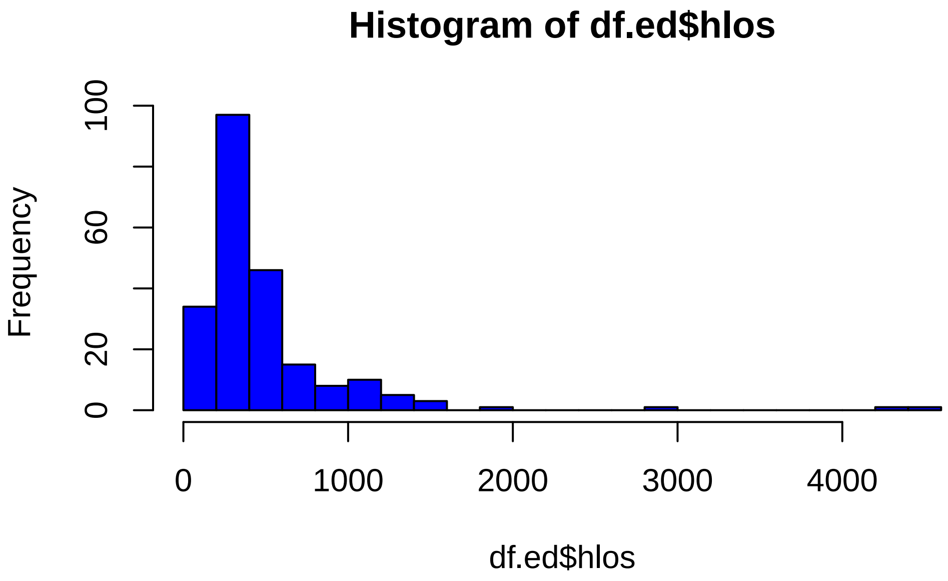
1.4 `gtsummary` for quick statistics

```r
library(gtsummary)

df.triage %>%
  select(temperature:dbp) %>%
  tbl_summary()
```

2 Checking Normality Visually

2.1 via Histograms

```r
df.ed$hlos <- as.numeric(df.ed$hlos)

hist(df.ed$hlos, col = "blue", breaks = 30)
```



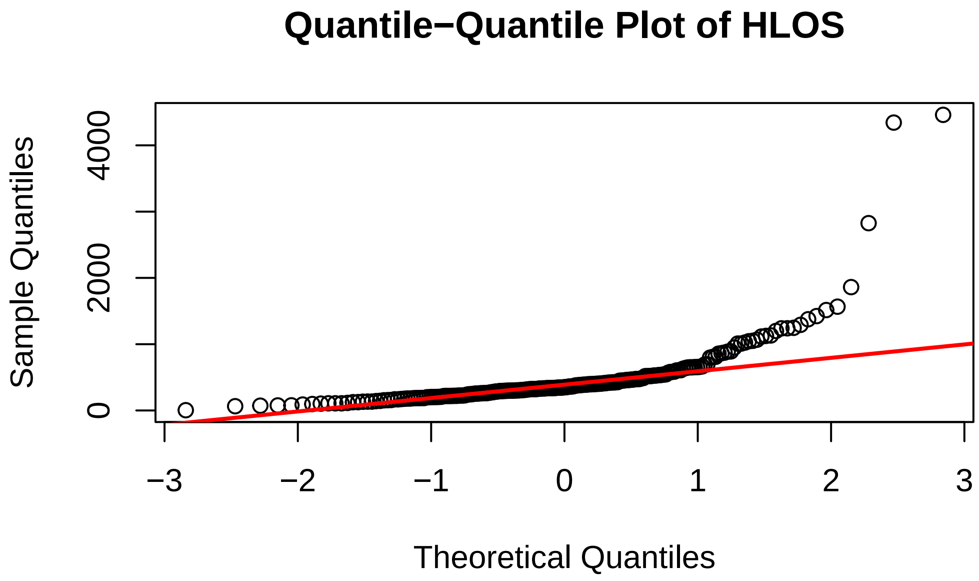
Widely used method

If the histogram is bell-shaped and symmetrical, it suggests normality

Dependent on the number of bins (breaks) and can be subjective

2.2 via Q-Q plots

```
qqnorm(df.ed$hlos, pch = 1, main = "Quantile-Quantile Plot of HLOS")
  qqline(df.ed$hlos, col = "red", lwd = 2)
```



Points should fall close to a straight diagonal line

Deviations from the line suggest departures from normality

- Points curving upwards: skewed to the right (positive skew)
- Points curving downwards: skewed to the left (negative skew)
- Points spread out more than the line: flatter than normal distribution
- Points bunched together closer than the line: heavier tails than normal distribution

3 Checking Normality Numerically

3.1 Shapiro-Wilk

```
library(stats)

swt <- shapiro.test(df.ed$hlos)
print(swt)
```

```

	Shapiro-Wilk normality test
```

```
data:  df.ed$hlos
W = 0.56083, p-value < 2.2e-16
```

Good for smaller samples

A p-value greater than 0.05 indicates that the data does not deviate significantly from the normal distribution

3.2 Kolmogorov-Smirnov

```r
library(stats)

kst <- ks.test(df.ed$hlos, "pnorm")
```

```
Warning in ks.test.default(df.ed$hlos, "pnorm"): ties should not be present for
the one-sample Kolmogorov-Smirnov test
```

```r
print(kst)
```

```


    Asymptotic one-sample Kolmogorov-Smirnov test

data:  df.ed$hlos
D = 0.99997, p-value < 2.2e-16
alternative hypothesis: two-sided
```

Good for larger samples

A p-value greater than 0.05 indicates that the data does not deviate significantly from the normal distribution

3.3 Skewness & Kurtosis

```r
library(e1071)

skewness <- skewness(df.ed$hlos)
kurtosis <- kurtosis(df.ed$hlos)

print(c("Skewness:", skewness, "Kurtosis:", kurtosis))
```

```
[1] "Skewness:"        "4.81774459323537" "Kurtosis:"        "31.1513455581506"
```

Kind of arbitrary, but these assess the asymmetry and tail heaviness of a distribution

Skewness:

- Absolute values close to 0 suggest symmetry
- Positive values indicate a right-skewed distribution
- Negative values indicate a left-skewed distribution

Kurtosis:

- A value of 3 indicates a normal distribution
- Values greater than 3 indicate heavier tails (leptokurtic)
- Values less than 3 indicate lighter tails (platykurtic)

4 Bivariate Analyses

4.1 Two continuous variables (1 dependent, 1 independent)

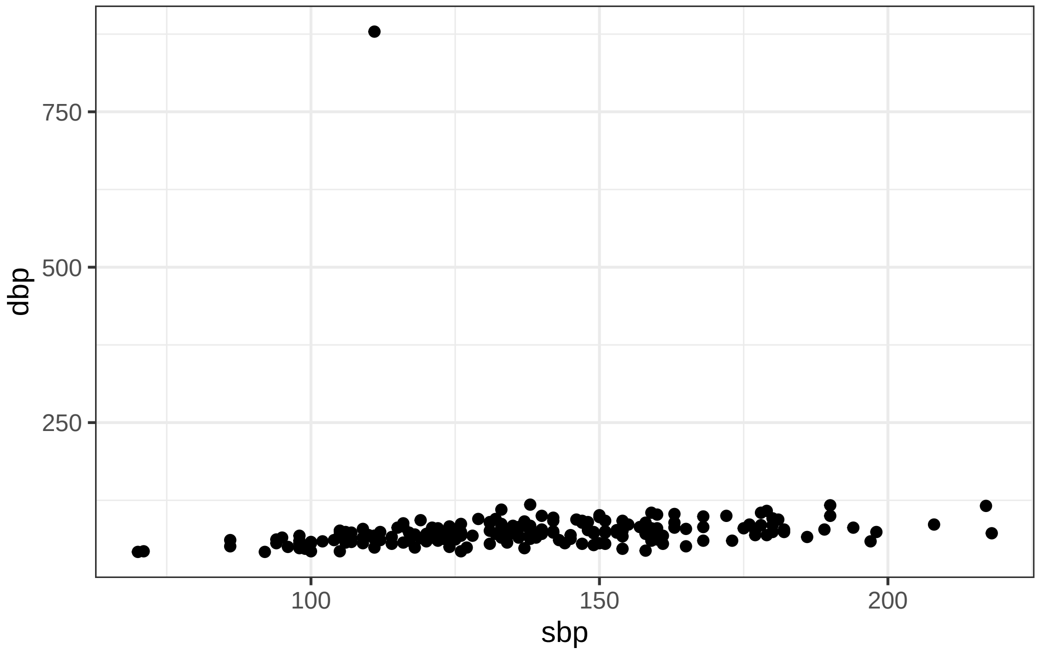
4.2 Scatter Plots

Two-variable scatterplots can be generated with `ggplot2` within the `tidyverse`

This is a highly customizable package that will be discussed in more detail later.

```r
library(ggplot2)

# Simple Plots
ggplot(df.triage, aes(sbp, dbp)) +
  geom_point() +
  theme_bw()
```

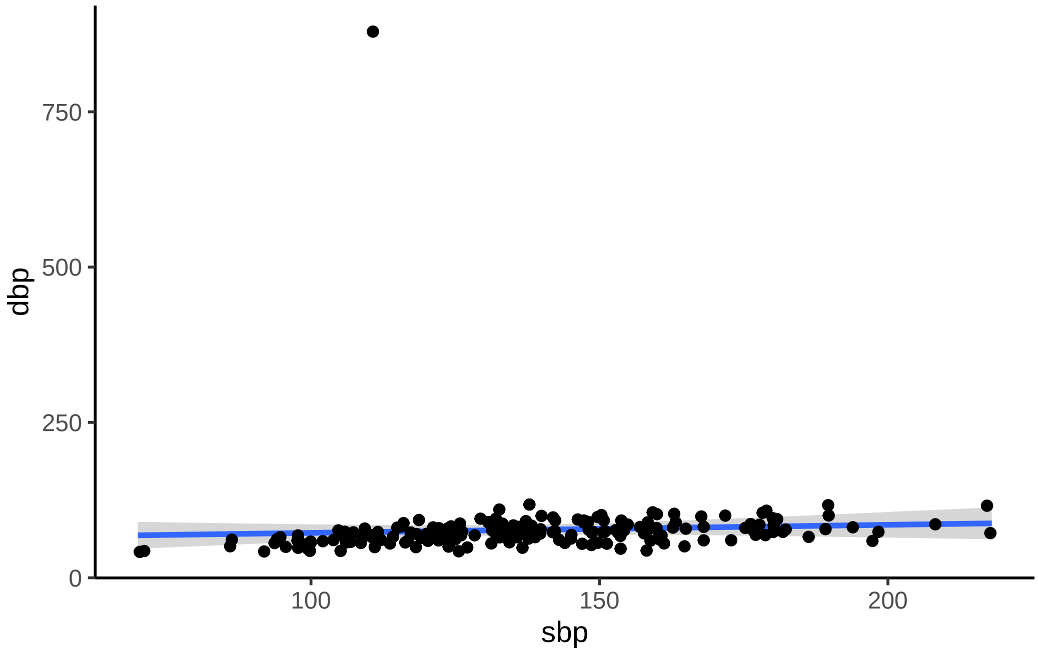
```
Warning: Removed 23 rows containing missing values or values outside the scale range
(`geom_point()`).
```



```
# With Regression Lines
ggplot(df.triage, aes(sbp, dbp)) +
  geom_smooth(method = "lm") +
  geom_jitter() +
  theme_classic()
```

```
`geom_smooth()` using formula = 'y ~ x'
```

```
Warning: Removed 23 rows containing non-finite outside the scale range
(`stat_smooth()`).
Removed 23 rows containing missing values or values outside the scale range
(`geom_point()`).
```



4.3 Pearson Correlation

```
# Calculate correlations
cor.test(df.triage$temperature, df.triage$sbp, method = "pearson")
```

```

	Pearson's product-moment correlation

data:  df.triage$temperature and df.triage$sbp
t = -0.22879, df = 194, p-value = 0.8193
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1562168  0.1240144
sample estimates:
        cor 
-0.01642376 
```

```
cor.test(~ temperature + sbp, data = df.triage, method = "pearson")
```

```
    Pearson's product-moment correlation

data:  temperature and sbp
t = -0.22879, df = 194, p-value = 0.8193
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1562168  0.1240144
sample estimates:
        cor 
-0.01642376 
```

the parametric method

The correlation_coefficient value will be between -1 and 1. - 1: Perfect positive correlation (variables increase/decrease together). - 0: No linear correlation (no relationship between variables). - -1: Perfect negative correlation (variables increase/decrease in opposite directions).

Correlation strength: - 0.0 to 0.2: Very weak or negligible - 0.2 to 0.4: Weak - 0.4 to 0.6: Moderate - 0.6 to 0.8: Strong - 0.8 to 1.0: Very strong

The p-value indicates the probability of observing the correlation by chance. Smaller p-values (usually < 0.05) suggest a statistically significant correlation.

4.4 Spearman Correlation

```r
# Calculate correlations
cor.test(df.triage$temperature, df.triage$sbp, method = "spearman")
```

```
Warning in cor.test.default(df.triage$temperature, df.triage$sbp, method =
"spearman"): Cannot compute exact p-value with ties
```

```
    Spearman's rank correlation rho

data:  df.triage$temperature and df.triage$sbp
S = 1158513, p-value = 0.2847
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho 
0.07680128 
```

```r
cor.test(~ temperature + sbp, data = df.triage, method = "spearman")
```

```
Warning in cor.test.default(x = mf[[1L]], y = mf[[2L]], ...): Cannot compute
exact p-value with ties
```

```

	Spearman's rank correlation rho

data:  temperature and sbp
S = 1158513, p-value = 0.2847
alternative hypothesis: true rho is not equal to 0
sample estimates:
       rho 
0.07680128 
```

The non-parametric method

It is suitable for measuring associations between variables measured on an ordinal scale (e.g., rankings, grades, levels).

It reflects the strength and direction of monotonic relationships, even if they are not perfectly linear.

Rho values range from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no correlation.

the p-value indicates the statistical significance of the correlation

4.5 Two variables (1 continuous dependent, 1 categorical independent)

4.6 Summary Statistics

`gtsummary` will provide the easiest and fastest method for quick bivariate statistics

```r
library(gtsummary)

df <- left_join(df.ed, df.triage, by = "stay_id")

df %>%
  select(race_white, temperature:dbp) %>%
  tbl_summary(by = "race_white")
```

Characteristic	**0** N = 80[1]	**1** N = 142[1]
temperature	98.00 (97.40, 98.20)	98.20 (97.70, 98.70)
Unknown	13	13
heartrate	84 (72, 96)	96 (80, 106)
Unknown	11	13
resprate	18 (16, 18)	18 (16, 20)
Unknown	11	12
o2sat	99 (97, 100)	98 (96, 99)
Unknown	11	13
sbp	142 (121, 163)	133 (116, 151)
Unknown	11	12
dbp	74 (61, 82)	72 (62, 83)
Unknown	11	12

[1] Median (Q1, Q3)

4.7 One-sample T-Test

```r
oneSamp.res <- t.test(df$age, mu = 75)
oneSamp.res
```

4.8 Two Samples

Equality of Variances - Levene's Test

```r
library(car)

lev.res1 <- leveneTest(age ~ inptDeath, data = df)
lev.res1

lev.res2 <- leveneTest(age ~ gender, data = df)
lev.res2
```

4.9 T-test for two groups

```
age.ttest.res <- t.test(age ~ inptDeath, data = df)
age.ttest.res
gend.ttest.res <- t.test(age ~ gender, data = df)
gend.ttest.res
```

4.10 Two-samples T-test

Requiring two separate dataframes with the same variables.

4.10.1 Unpaired Equal Variance

```
# subset to get our "two samples"
df.white <- subset(df, race_white == 1)
df.nonwhite <- subset(df, race_white == 0)

# Welch's T-Test, equal variance assumed
twoSampEq.res <- t.test(df.white$hlos, df.nonwhite$hlos, paired = FALSE)
twoSampEq.res
```

```

    Welch Two Sample t-test

data:  df.white$hlos and df.nonwhite$hlos
t = -1.0625, df = 131.26, p-value = 0.29
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -233.30843   70.25974
sample estimates:
mean of x mean of y
 456.4021  537.9265
```

4.10.2 Unpaired Unequal Variance

```
# Welch's T-Test, unequal variance
twoSampUnEq.res <- t.test(df.white$hlos, df.nonwhite$hlos, paired = FALSE, var.equal = FALSE)
twoSampUnEq.res
```

```
    Welch Two Sample t-test

data:  df.white$hlos and df.nonwhite$hlos
t = -1.0625, df = 131.26, p-value = 0.29
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -233.30843   70.25974
sample estimates:
mean of x mean of y
 456.4021  537.9265
```

4.11 Paired t-test

Requiring one data frame with two columns to be compared.

```
# Paired t-test
paired.res <- t.test(df$baselineMeasure, df$followupMeasure, paired = TRUE)
paired.res
```

5 Analysis of Continuous Data w/ Polychotomous Categorical

5.1 ANOVA

5.2 Equality of Variance Test

```
# Bartlett's Test
age.bartl.res <- bartlett.test(age ~ admission_type, data = df)
age.bartl.res
```

`age` is set as the "dependent variable" and is continuous

`admission_type` is the "independent variable" and is categorical w/ more than 2 categories.

5.3 One-way ANOVA

```
one.way <- aov(hlos ~ admission_type, data = df)
summary(one.way)
head(one.way)
```

5.4 Two-way ANOVA [Multivariate!]

```
two.way <- aov(hlos ~ admission_type + race, data = df)
summary(two.way)
```

5.5 Test of Medians

```
# Mann-Whitney U / Wilcoxon Rank Sum Test
  # two groups
wilcox <- wilcox.test(hlos ~ race, data = df)
wilcox

# Kruskal-Wallis Test
  # more than two groups
krusk <- kruskal.test(hlos ~ admission_type, data = df)
krusk
```