Four Statistical Tests and Four Statistical Framworks

EA030

2025-10-01

Test for Assocation

Example #1: Polluters and Mobility

The third test is the Pearson correlation test. This test is used to test the association between two or more categories of count data.

Example #2: Polluters and Residential Mobility

```
##            Polluter Non-polluter
## Non-mobile      112           49
## Mobile          130           85
```

```
chisq.test(bubbles)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  bubbles
## X-squared = 2.9388, df = 1, p-value = 0.08647
```

Example #3: Cattle Grazing and Water Quality

Water quality is a key concern in areas where cattle grazing overlaps with recreational use. Previous studies (Derlet et al., 2008, *Water Quality Conditions Associated with Cattle Grazing and Recreation on National Forest Lands*) have shown that cattle presence is associated with elevated levels of *Escherichia coli* (E. coli), an indicator of fecal contamination.

This analysis tests whether there is a statistically significant association between the presence of cattle and *E. coli* detection in water samples.

Data The following hypothetical dataset reflects water samples collected in two conditions: **sites with cattle present** and **sites without cattle**.

Cattle Presence	E. coli Detected	E. coli Not Detected	Total
Yes	28	12	40
No	10	30	40
Total	38	42	80

```r
# Create contingency table
data <- matrix(c(28, 12,
                 10, 30),
               nrow = 2, byrow = TRUE)

colnames(data) <- c("Ecoli_Detected", "Ecoli_NotDetected")
rownames(data) <- c("Cattle_Present", "No_Cattle")
data <- as.table(data)

# Print table
data
```

R Code

```
##                  Ecoli_Detected Ecoli_NotDetected
## Cattle_Present             28                12
## No_Cattle                  10                30
```

```r
# Perform Chi-square test
chisq.test(data)
```

```
## 
##  Pearson's Chi-squared test with Yates' continuity correction
## 
## data:  data
## X-squared = 14.486, df = 1, p-value = 0.0001412
```

Regression Analysis

Fluorescence sensors provide a real-time, continuous monitoring method for detecting dissolved organic matter and microbial contamination in aquatic environments.
In the study by Griffith et al. (2009), *Evaluation of real-time fluorescence sensors and benchtop fluorescence for tracking and predicting sewage contamination in the Tijuana River Estuary at the US-Mexico border*, regression analysis was applied to test whether fluorescence intensity could predict bacterial contamination indicators (e.g., *E. coli*, Enterococcus).

Here, we construct a regression model using simulated but realistic data to illustrate how fluorescence intensity relates to *E. coli* concentration.

Example #4 Testing Reliability of a Method

Below is a simulated dataset representing paired measurements of fluorescence intensity (arbitrary units) and *E. coli* concentrations (log CFU/100 mL).

Sample	Fluorescence_Intensity	Ecoli_LogCFU
1	15.2	2.1
2	18.5	2.4
3	22.3	2.9
4	25.1	3.0

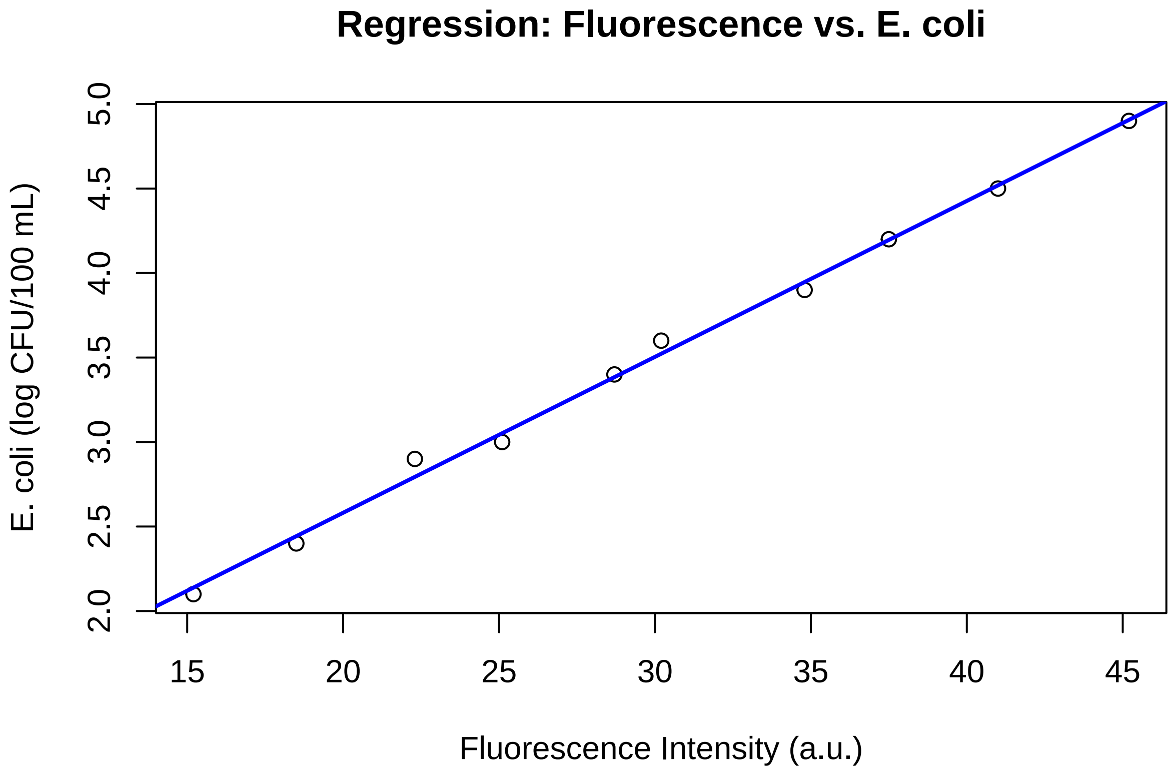
Sample	Fluorescence_Intensity	Ecoli_LogCFU
5	28.7	3.4
6	30.2	3.6
7	34.8	3.9
8	37.5	4.2
9	41.0	4.5
10	45.2	4.9

R Code

```r
# Create dataset
fluorescence <- c(15.2, 18.5, 22.3, 25.1, 28.7, 30.2, 34.8, 37.5, 41.0, 45.2)
ecoli <- c(2.1, 2.4, 2.9, 3.0, 3.4, 3.6, 3.9, 4.2, 4.5, 4.9)

data <- data.frame(Fluorescence_Intensity = fluorescence,
                   Ecoli_LogCFU = ecoli)

# Fit linear regression model
model <- lm(Ecoli_LogCFU ~ Fluorescence_Intensity, data = data)

# Summary of model
summary(model)
```

```
##
## Call:
## lm(formula = Ecoli_LogCFU ~ Fluorescence_Intensity, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05178 -0.04179 -0.01239  0.01313  0.10653
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            0.736164   0.060791   12.11  2.0e-06 ***
## Fluorescence_Intensity 0.092256   0.001944   47.45  4.3e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05726 on 8 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.996
## F-statistic:  2252 on 1 and 8 DF,  p-value: 4.301e-11
```

```r
# Plot relationship
plot(data$Fluorescence_Intensity, data$Ecoli_LogCFU,
     xlab = "Fluorescence Intensity (a.u.)",
     ylab = "E. coli (log CFU/100 mL)",
     main = "Regression: Fluorescence vs. E. coli")
abline(model, col = "blue", lwd = 2)
```

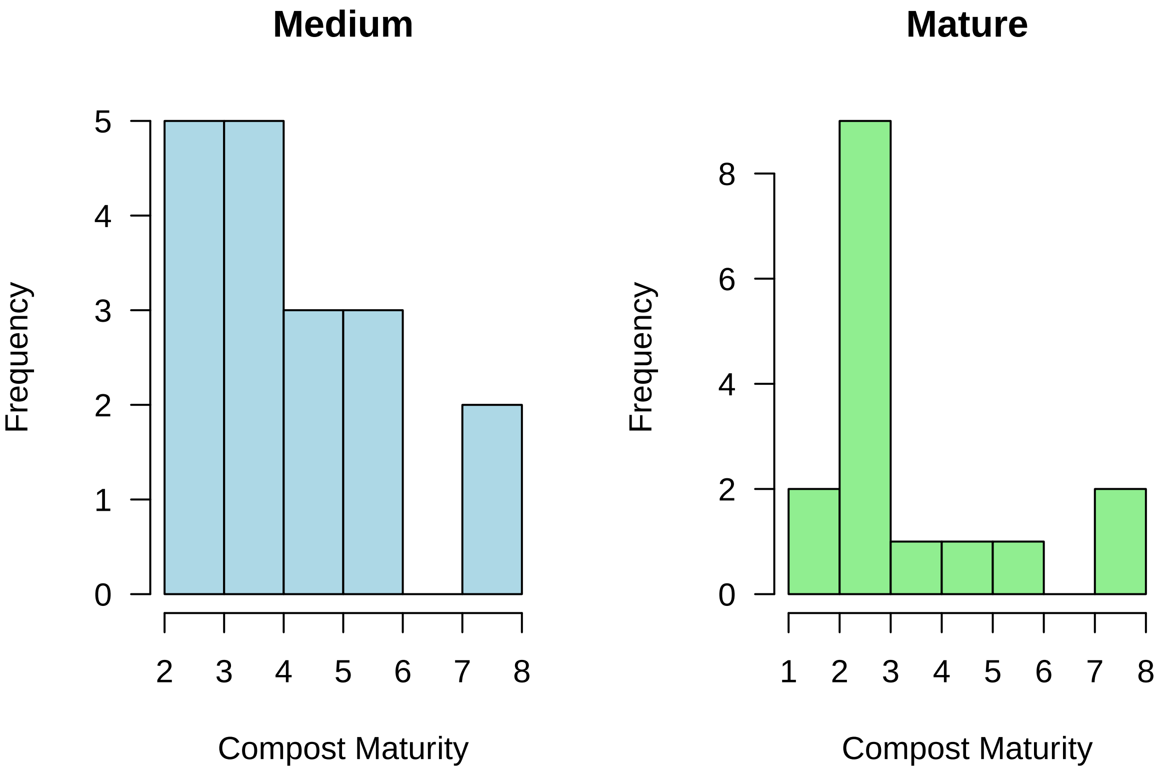


Testing for Population Differences: t-test

Example #4: Compost Maturity

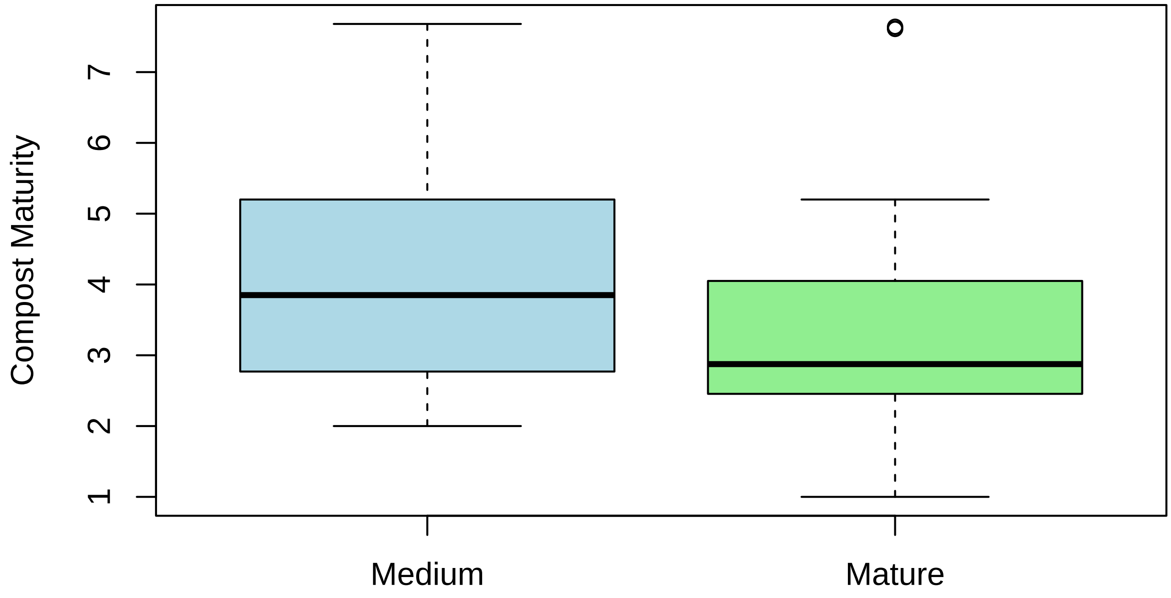
The second test is the paired t-test. This test is used to compare the means of two groups.

```
medium = c(4, 2, 3.25, 3.25, 2.55, 2.34, 3.7, 3.7, 2.69, 2.77, 4.3, 5.2, 7.63, 7.68, 6, 6, 4.7, 4.5)
mature = c(2, 1, 2.65, 2.95, 2.26, 2.12, 5, 5.2, 2.85, 2.69, 3.1, 2.8, 7.64, 7.61, 3, 2.9)

par(mfrow=c(1,2), las=1)
hist(medium, main="Medium", xlab="Compost Maturity", col="lightblue")
hist(mature, main="Mature", xlab="Compost Maturity", col="lightgreen")
```



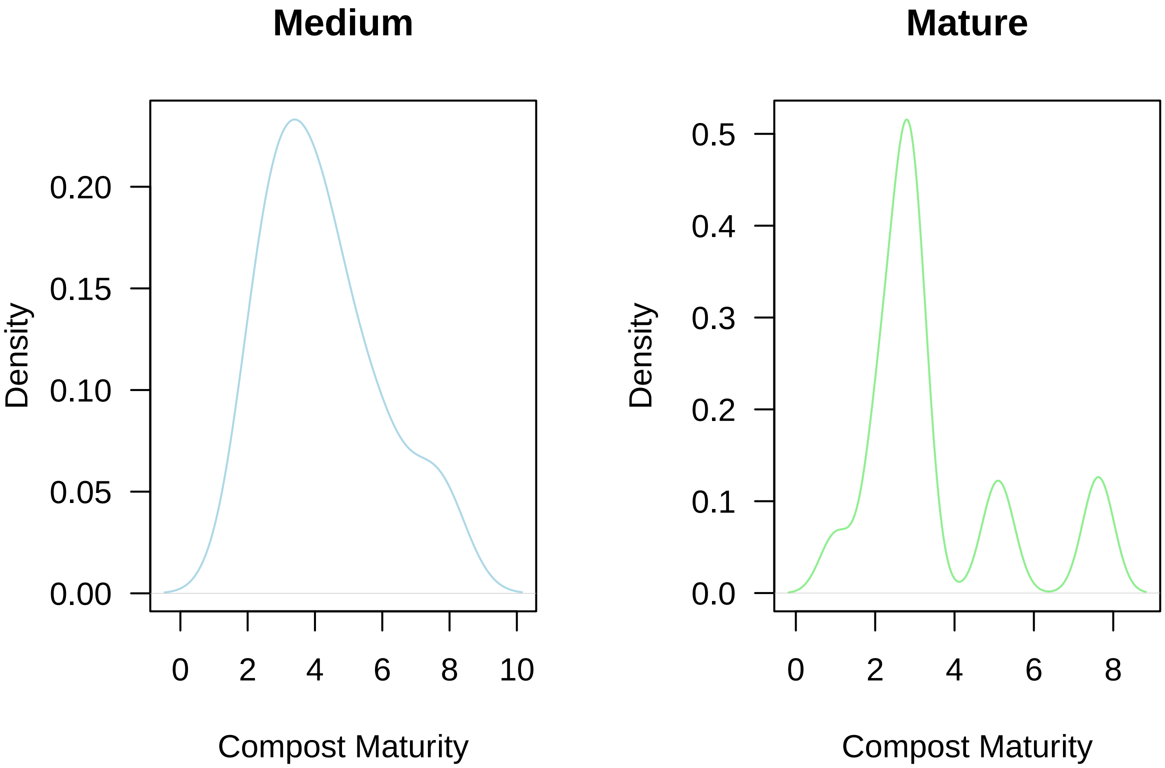
showing these in a density distribution and variablity Boxplots are good ways of showing "categorical predictors"

```
boxplot(medium, mature, names=c("Medium", "Mature"),
        col=c("lightblue", "lightgreen"), ylab="Compost Maturity")
```



This is a bit of a sublety...

```
par(mfrow=c(1,2), las=1)
plot(density(medium), main="Medium", xlab="Compost Maturity", col="lightblue")
plot(density(mature), main="Mature", xlab="Compost Maturity", col="lightgreen")
```



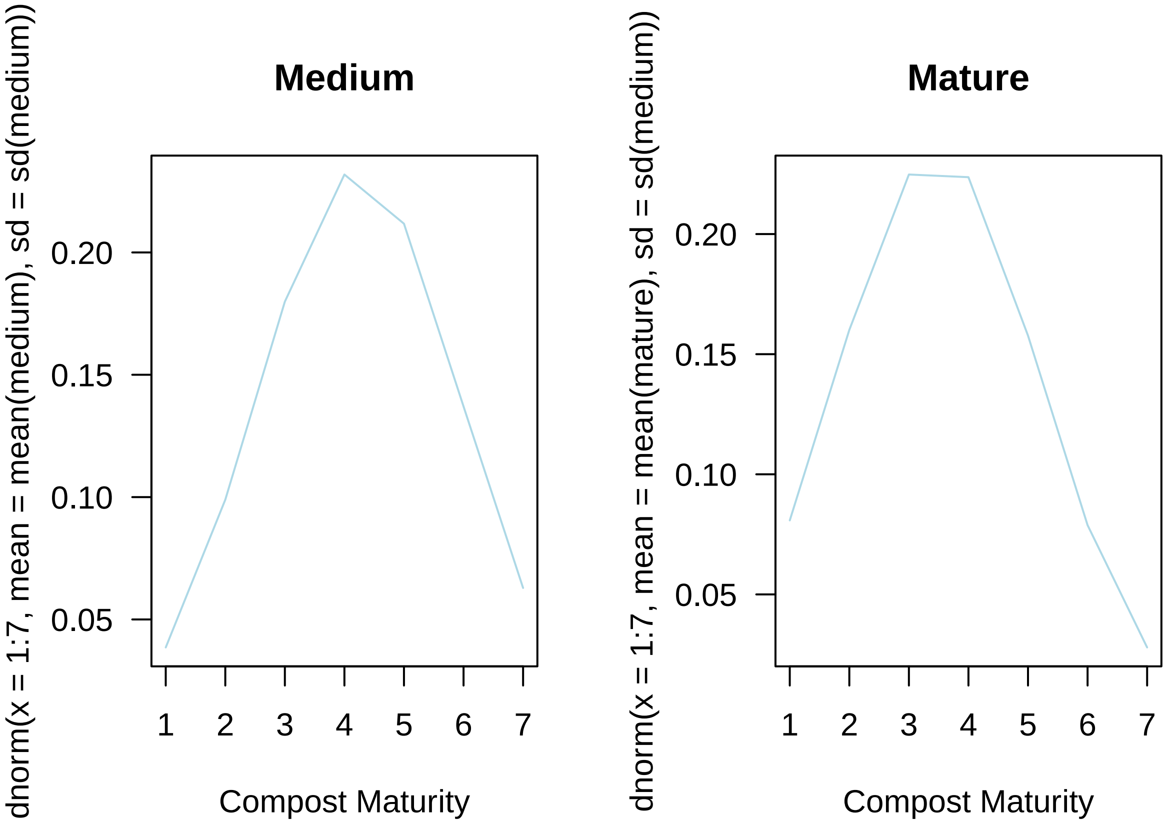
```r
t.test(medium, mature, paired=FALSE)
```

R function for t-test

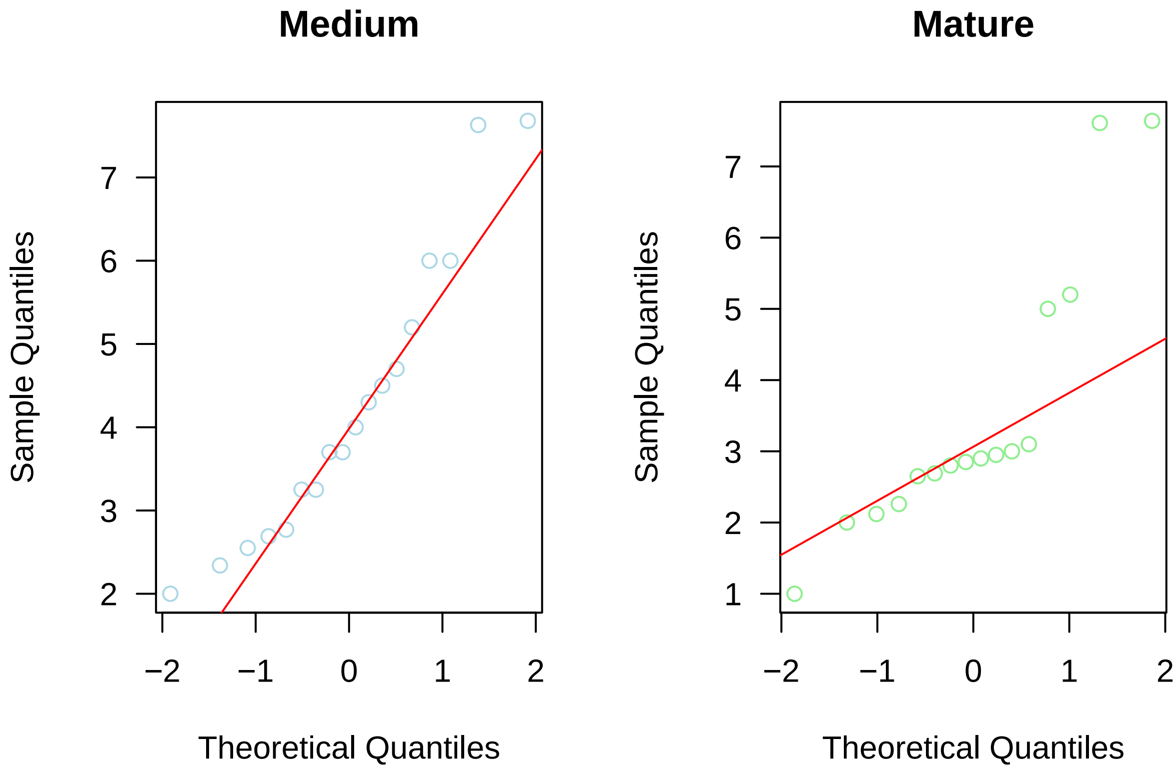
```
##
##  Welch Two Sample t-test
##
## data:  medium and mature
## t = 1.2052, df = 30.366, p-value = 0.2374
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5209723  2.0230557
## sample estimates:
## mean of x mean of y
##  4.236667  3.485625
```

Ploting the data using a normal distribution Plotting Distributions – why might this be useful?

```r
par(mfrow=c(1,2), las=1)
plot(dnorm(x=1:7, mean=mean(medium), sd=sd(medium)), main="Medium", xlab="Compost Maturity", ty="l", co
plot(dnorm(x=1:7, mean=mean(mature), sd=sd(medium)), main="Mature", xlab="Compost Maturity", ty="l", co
```



```
qqnorm(medium, main="Medium", col="lightblue")
qqline(medium, col="red")
qqnorm(mature, main="Mature", col="lightgreen")
qqline(mature, col="red")
```



ANOVA

For this exercise, you will using the following data sets to analyze and interpret the results of four statistical tests and four statistical frameworks.

Example #5: Testing if Treatment Means are Equal

The first test is the Analysis of Variances (ANOVA) test. This test is used to compare the means of three or more groups.

```
treatments = c("A", "B", "C", "D")
```

Creating a Dataset The data set tests the treatments of soil restoration in 'The Wash' and includes 10 replicated measurements for each treatment.

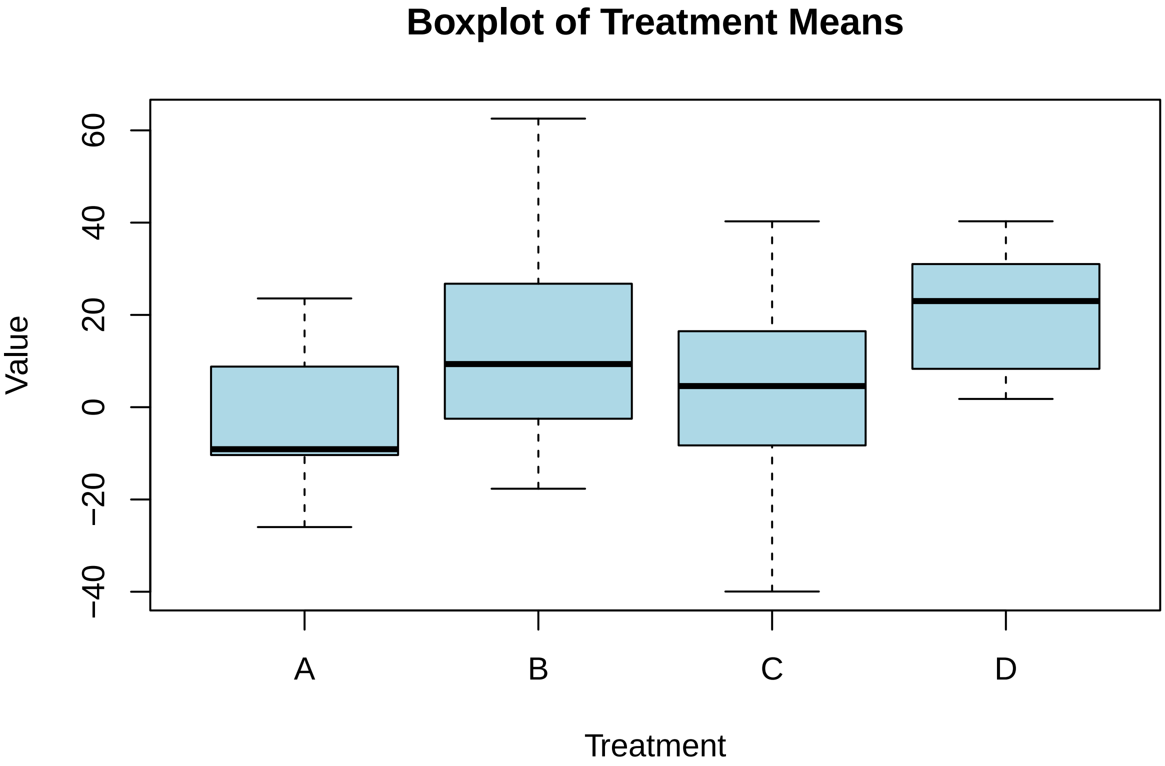
To create the data set, use the following code that defines the treatments and generates the data for each treatment. The data is generated using the `rnorm` function. The `rnorm` function generates random numbers from a normal distribution. The `mean` and `sd` parameters are used to specify the mean and standard deviation of the normal distribution. The `rnorm` function is used to generate 10 random numbers for each treatment. The mean and standard deviation of the normal distribution are set to 10 and 2, respectively.

```
print(xtable(cbind(replicates, a, b, c, d)), type="latex")
```

% latex table generated in R 4.2.2 by xtable 1.8-4 package % Wed Oct 1 10:40:27 2025

Boxplot The boxplot is used to visualize the data (simulated data).

```r
boxplot(Value ~ Treatment, data = anovadata, col = "lightblue",
        main = "Boxplot of Treatment Means",
        xlab = "Treatment", ylab = "Value")
```



Testing the Assumptions of ANOVA The first step in the ANOVA test is to test the assumptions of the test. The assumptions of the ANOVA test are that the data is normally distributed and that the variances of the groups are equal.

The normality of the data is tested using the Shapiro-Wilk test. The Shapiro-Wilk test is used to test the null hypothesis that the data is normally distributed. The alternative hypothesis is that the data is not normally distributed.

```r
summary(aov(Value ~ Treatment, data = anovadata))
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Treatment    3   3132  1043.8   2.793 0.0542 .
## Residuals   36  13452   373.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The null hypothesis is that the means of the three species are equal. The alternative hypothesis is that the means of the three species are not equal. The ANOVA test is used to test the null hypothesis.

ANOVA Example #6: Arsenic Concentrations in Unregulated Water Sources

Elevated arsenic and uranium concentrations in unregulated water sources pose major public health challenges. Hoover et al. (2017), *Elevated Arsenic and Uranium Concentrations in Unregulated Water Sources on the Navajo Nation, USA*, highlighted that contaminant levels can differ across water sources such as wells, springs, and livestock tanks.

Here, we use a **one-way ANOVA** to test whether **arsenic concentrations differ significantly by water source type**.

Data Simulated arsenic concentration data (in µg/L) across three source types:

Sample	SourceType	Arsenic
1	Well	35.2
2	Well	41.0
3	Well	38.7
4	Spring	12.5
5	Spring	15.2
6	Spring	10.8
7	LivestockTank	55.1
8	LivestockTank	61.3
9	LivestockTank	58.9

```
# Simulated dataset
SourceType <- factor(c("Well","Well","Well",
                       "Spring","Spring","Spring",
                       "LivestockTank","LivestockTank","LivestockTank"))

Arsenic <- c(35.2,41.0,38.7,
             12.5,15.2,10.8,
             55.1,61.3,58.9)

data <- data.frame(SourceType, Arsenic)

# Run one-way ANOVA
anova_model <- aov(Arsenic ~ SourceType, data = data)

# Show results
summary(anova_model)
```

R Code

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## SourceType   2 3133.3  1566.6   202.3 3.12e-06 ***
## Residuals    6   46.5     7.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Post-hoc pairwise comparisons
TukeyHSD(anova_model)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Arsenic ~ SourceType, data = data)
##
```

```
## $SourceType
##                            diff       lwr       upr     p adj
## Spring-LivestockTank -45.60000 -52.57078 -38.62922 0.0000023
## Well-LivestockTank   -20.13333 -27.10411 -13.16255 0.0002822
## Well-Spring           25.46667  18.49589  32.43745 0.0000743
```

Logistic Regression

The third test is the logistic regression test. This test is used to test the association between a binary response variable and one or more predictor variables.

Example #7: Distance and Success

Small square in tape on floor, 10x10 cm, between 20-300 cm, try to get in the square, by rolling or sliding the socket extension.

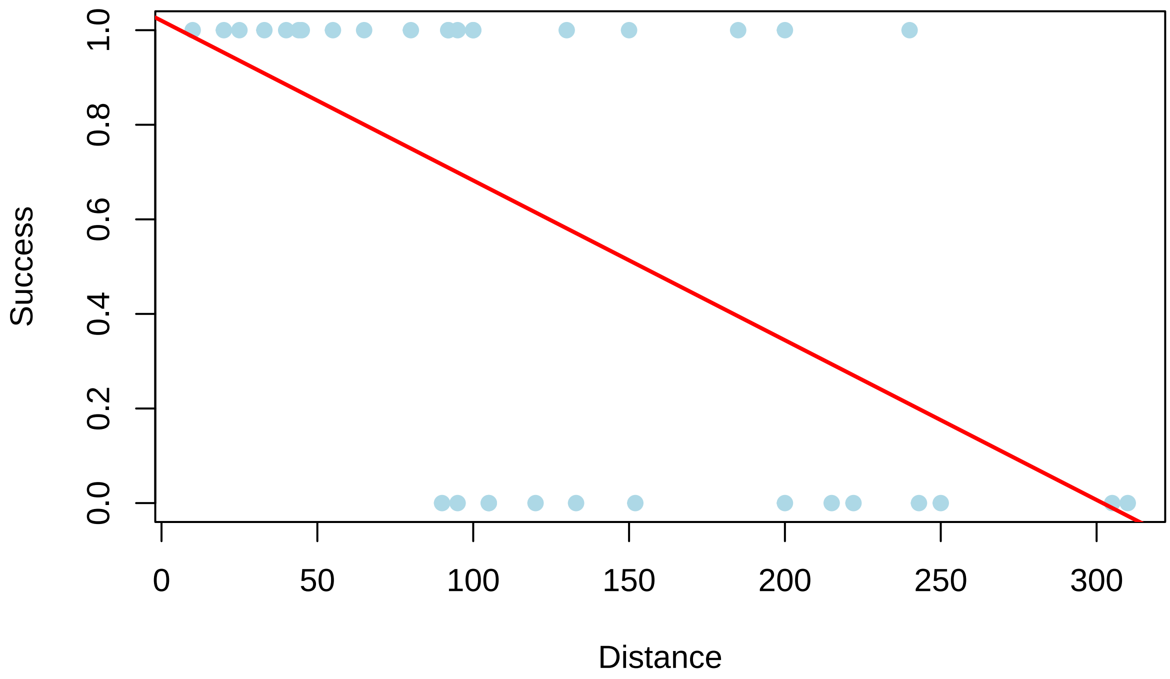
```
##    Distance Success logical
## 8        10       1    TRUE
## 9        20       1    TRUE
## 26       25       1    TRUE
## 27       33       1    TRUE
## 10       40       1    TRUE
## 28       44       1    TRUE
```

```
str(mydata)
```

Plotting Data – Linear Model

```
## 'data.frame':    34 obs. of  3 variables:
##  $ Distance: num  10 20 25 33 40 44 45 55 65 80 ...
##  $ Success : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ logical : logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
```

```
plot(mydata$Distance, mydata$Success, col="lightblue", pch=19,
     ylab="Success", xlab="Distance")
abline(coef(lm(Success ~ Distance, data = mydata)), col="red", lwd=2)
```



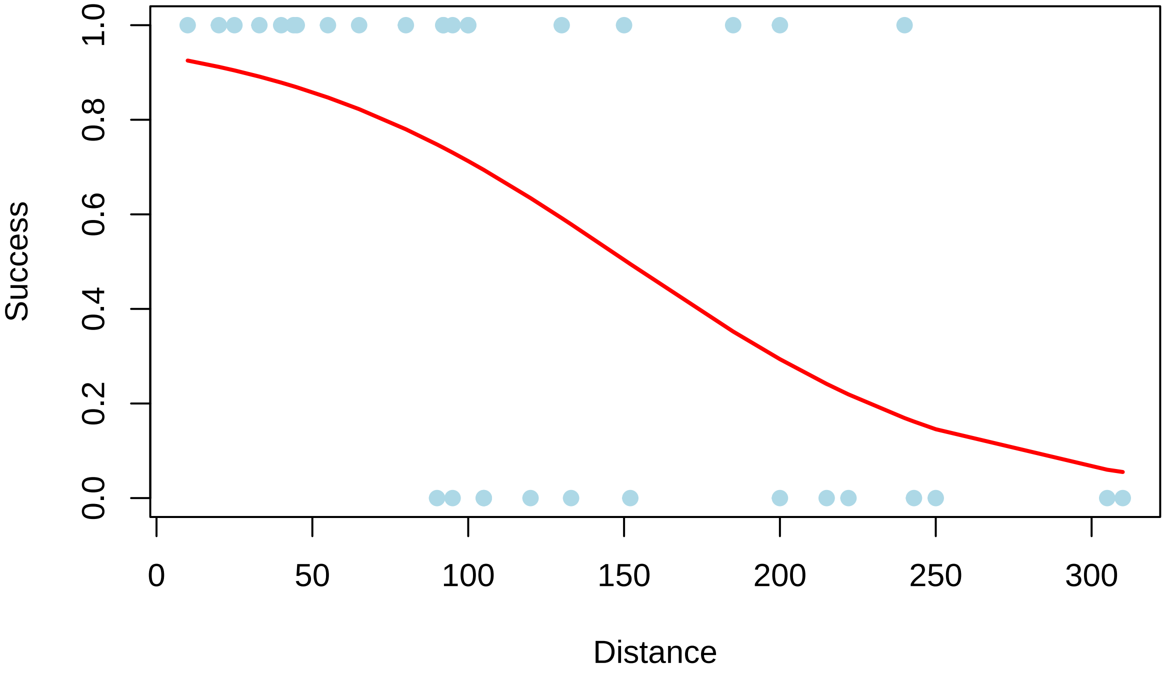
```
mylogit <- glm(logical ~ Distance, data = mydata, family = "binomial")
summary(mylogit)
```

Logistc Model

```
##
## Call:
## glm(formula = logical ~ Distance, family = "binomial", data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6597  -0.7334   0.4640   0.7744   1.8851
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.693238   0.927043   2.905  0.00367 **
## Distance    -0.017853   0.006391  -2.793  0.00522 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 46.070  on 33  degrees of freedom
## Residual deviance: 34.371  on 32  degrees of freedom
## AIC: 38.371
##
## Number of Fisher Scoring iterations: 4
```

Plot Results

```
plot(mydata$Distance, mydata$logical, col="lightblue", pch=19, ylab=c("Success"), xlab=c("Distance"), yl
lines(mydata$Distance, fitted(mylogit), col="red", lwd=2)
```



Example #8: Logistic Regression and Structural Violence in the Tijuana River

Vulnerable populations living in the Tijuana River canal face multiple forms of structural violence, including flooding risk, sewage exposure, and police harassment.
Vogt (2018), *Deported, homeless, and into the canal: Environmental structural violence in the binational Tijuana River*, highlights how deported and homeless individuals are disproportionately exposed to hazardous environments.

Here, we demonstrate the use of **logistic regression** to examine whether social and environmental factors (homelessness, deportation status, and proximity to the canal) predict the probability of **high exposure to risk**.

Data The table below is a simulated dataset of 20 individuals. The outcome variable (`HighRisk`) is binary (1 = high environmental exposure risk, 0 = low risk). Predictors include:
- `Homeless` (1 = yes, 0 = no)
- `Deported` (1 = yes, 0 = no)
- `NearCanal` (Distance in Meters)

ID	Homeless	Deported	NearCanal	HighRisk
1	1	1	2	1
2	1	1	4	1
3	1	0	3	1
4	1	1	26	1
5	0	1	1.3	1
6	0	1	30	0
7	1	0	42	0
8	0	0	2.1	0
9	0	0	4	0
10	1	1	2	1

ID	Homeless	Deported	NearCanal	HighRisk
11	0	1	5	1
12	1	0	2	1
13	1	1	59	1
14	0	0	80	0
15	0	0	3	0
16	1	1	7	1
17	0	1	12	0
18	1	0	19	0
19	0	1	14	1
20	1	1	2	1

```r
# Simulated dataset
ID <- 1:20
Homeless <- c(1,1,1,1,0,0,1,0,0,1,0,1,1,0,0,1,0,1,0,1)
# Deported <- c(1,1,0,1,1,1,1,0,0,0,1,1,0,1,0,0,1,1,0,1,1)
NearCanal <- c(2,4,3,26,1.3,30,42,2.1,4.0,2,5,2,50,80,3,13,12,29,24,2)
HighRisk <- c(1,1,1,0,1,0,0,1,0,1,1,1,1,0,0,1,0,0,0,1)

data <- data.frame(ID, Homeless, NearCanal, HighRisk)
data <- data[order(data$NearCanal),]
```

R Code

```r
# Logistic regression
model <- glm(HighRisk ~ NearCanal,
             family = binomial(link="logit"), data = data)

summary(model)
```

Data Analysis

```
##
## Call:
## glm(formula = HighRisk ~ NearCanal, family = binomial(link = "logit"),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6436  -0.8876   0.7430   0.7586   2.1733
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   1.2623     0.7106   1.776   0.0757 .
## NearCanal    -0.0705     0.0382  -1.846   0.0649 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

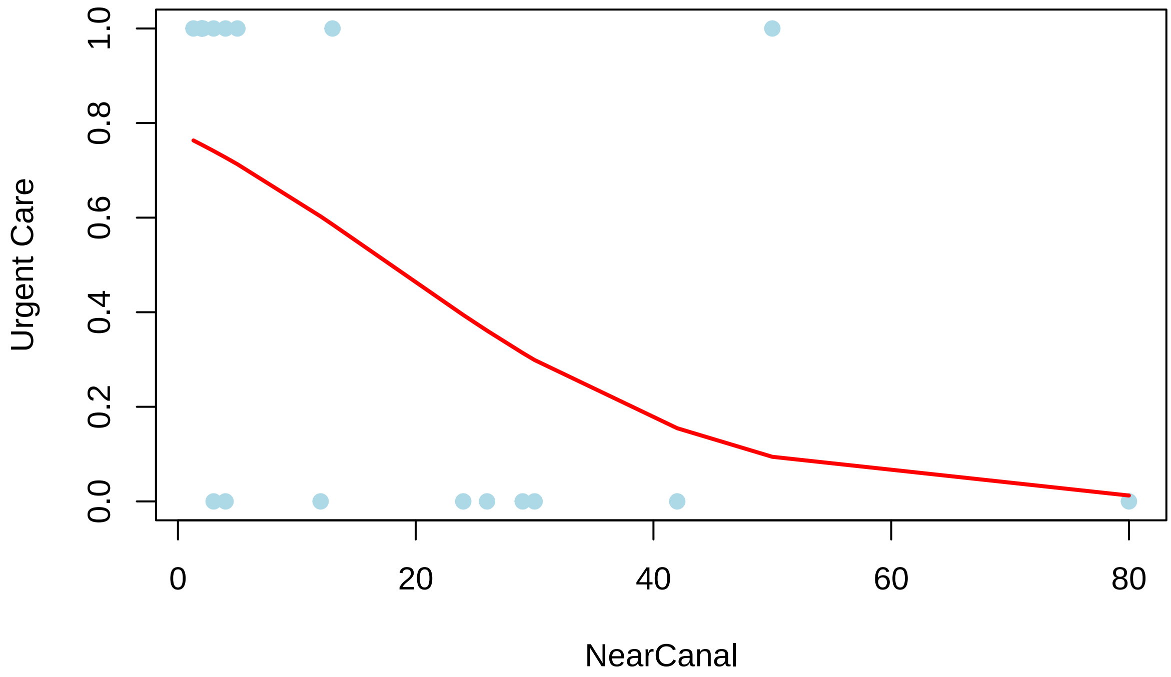
```
##
##     Null deviance: 27.526  on 19  degrees of freedom
## Residual deviance: 21.939  on 18  degrees of freedom
## AIC: 25.939
##
## Number of Fisher Scoring iterations: 5
```

```
# Convert to odds ratios
exp(cbind(OR = coef(model), confint(model)))
```

```
## Waiting for profiling to be done...
```

```
##                    OR     2.5 %    97.5 %
## (Intercept) 3.5336730 0.9771926 17.279631
## NearCanal   0.9319303 0.8500544  0.990163
```

```
plot(data$NearCanal, data$HighRisk, col="lightblue", pch=19, ylab=c("Urgent Care"), xlab=c("NearCanal")
lines(data$NearCanal, fitted(model), col="red", lwd=2)
```



	replicates	a	b	c	d
1	1.00	-9.59	8.71	40.28	22.08
2	2.00	8.80	-7.21	-12.34	31.02
3	3.00	6.53	12.92	-39.95	26.85
4	4.00	17.57	62.54	8.01	40.29
5	5.00	-20.56	-17.67	-8.28	7.30
6	6.00	-10.37	3.97	-2.20	31.26
7	7.00	-25.98	26.76	1.15	8.31
8	8.00	23.57	9.98	15.04	14.01
9	9.00	-8.62	36.67	32.14	23.92
10	10.00	-10.27	-2.50	16.47	1.79
11	1.00	-9.59	8.71	40.28	22.08
12	2.00	8.80	-7.21	-12.34	31.02
13	3.00	6.53	12.92	-39.95	26.85
14	4.00	17.57	62.54	8.01	40.29
15	5.00	-20.56	-17.67	-8.28	7.30
16	6.00	-10.37	3.97	-2.20	31.26
17	7.00	-25.98	26.76	1.15	8.31
18	8.00	23.57	9.98	15.04	14.01
19	9.00	-8.62	36.67	32.14	23.92
20	10.00	-10.27	-2.50	16.47	1.79
21	1.00	-9.59	8.71	40.28	22.08
22	2.00	8.80	-7.21	-12.34	31.02
23	3.00	6.53	12.92	-39.95	26.85
24	4.00	17.57	62.54	8.01	40.29
25	5.00	-20.56	-17.67	-8.28	7.30
26	6.00	-10.37	3.97	-2.20	31.26
27	7.00	-25.98	26.76	1.15	8.31
28	8.00	23.57	9.98	15.04	14.01
29	9.00	-8.62	36.67	32.14	23.92
30	10.00	-10.27	-2.50	16.47	1.79
31	1.00	-9.59	8.71	40.28	22.08
32	2.00	8.80	-7.21	-12.34	31.02
33	3.00	6.53	12.92	-39.95	26.85
34	4.00	17.57	62.54	8.01	40.29
35	5.00	-20.56	-17.67	-8.28	7.30
36	6.00	-10.37	3.97	-2.20	31.26
37	7.00	-25.98	26.76	1.15	8.31
38	8.00	23.57	9.98	15.04	14.01
39	9.00	-8.62	36.67	32.14	23.92
40	10.00	-10.27	-2.50	16.47	1.79