

# 371 R Functions

This is a list of all of the important R functions we use in 371. I'll be updating this as the semester progresses. If there is a function missing that you believe should be on this list, let me know.

I've included quick examples of each function being used. The outputs are **not** shown in this document, but you can copy/paste the examples to run them yourself.

## Notes 2 New R Functions

`c()` - combine values into a data vector

```
milkfat <- c(20, 38, 44, 49, 42, 21, 48, 47, 21,  
            22, 23, 26, 26, 44, 48, 29)
```

`hist()` - create a histogram of a data vector

```
hist(milkfat, main = "Histogram of Milk Fat Data",  
     col = "dodgerblue", breaks = seq(20, 50, 5))
```

`median()` - find the median value of a data vector

```
median(milkfat)
```

`sum()` - add all of the values in a data vector

```
vals <- c(1, 3, 6, 2)  
sum(vals)
```

`length()` - find the length (number of values) of a data vector

```
length(milkfat)
```

`mean()` - find the average value of a data vector

```
mean(milkfat)
```

`quantile()` - find any quantile/percentile of a data vector

```
quantile(milkfat, probs = c(0.2, 0.5, 0.8))
```

`IQR()` - find the interquartile range of a data vector

```
IQR(milkfat)
```

`range()` - find the range (min and max) of a data vector

```
range(milkfat)
```

`sd()` - find the standard deviation of a data vector

```
sd(milkfat)
```

`var()` - find the variance of a data vector

```
var(milkfat)
```

`sqrt()` - find the square root of a number

```
sqrt(9)
```

boxplot() - create a boxplot of one or more data vectors

```
milkfat_supp <- milkfat
milkfat_nosupp <- c(20, 21, 22, 22, 23, 24, 18, 27, 25,
                   25, 19, 21, 28, 18, 17, 21, 23)

boxplot(milkfat_supp, milkfat_nosupp, main = "Milk Fat",
        names = c("Supplement", "No Supplement"))
```

## Notes 4 New R Functions

dbinom() - find a specific probability of a binomial distribution

```
dbinom(2, size = 3, prob = 9/16)

dbinom(1:3, size = 3, prob = 9/16)
sum(dbinom(1:3, size = 3, prob = 9/16))
```

pnorm() - find the area (probability) to the left of a value on a normal distribution

```
pnorm(2/3)
1 - pnorm(2/3)
pnorm(2/3, lower.tail = F)

pnorm(72, mean = 70, sd = 3)

pnorm(16, mean = 16.1, sd = 0.3, lower.tail = F)
```

qnorm() - find a specific quantile/percentile of a normal distribution

```
qnorm(0.5)
qnorm(0.975)

qnorm(0.99, 16.1, 0.3)
qnorm(0.99)
qnorm(0.99)*0.3 + 16.1
```

## Notes 5 New R Functions

rnorm() - draw a random sample from a normal population

```
rnorm(15)

rnorm(20, mean = 70, sd = 3)
```

qqnorm() - create a normal qq-plot of a data vector

```
x <- rnorm(15)
qqnorm(x)

qqnorm(milkfat)
```

## Notes 6 New R Functions

qt() - find a specific quantile/percentile of a t distribution

```
qt(0.975, df = 39)
qt(0.025, df = 39)
```

```
qt(0.975, df = 5)
qt(0.975, df = 10)
qt(0.975, df = 15)
```

`t.test()` - automatically build a t CI (and perform a t hypothesis test) with a specified level

```
seedlings <- c(2.6, 1.9, 1.8, 1.6, 1.4, 2.2, 1.2, 1.6, 1.6,
              1.5, 1.4, 1.6, 2.3, 1.5, 1.1, 1.6, 2.0, 1.5,
              1.7, 1.5, 1.6, 2.1, 2.2, 1.0, 1.2, 1.2, 1.8,
              1.7, 0.8, 1.5, 2.0, 2.2, 1.5, 1.6, 2.2, 2.1,
              1.6, 1.7, 1.7, 1.2)
```

```
t.test(seedlings, conf.level = 0.95)
```

## Notes 7 New R Functions

`pt()` - find the area (probability) to the left of a value on a t distribution

```
pt(2, df = 10)
```

```
2*pt(2.664, df = 39, lower.tail = F)
```

## Notes 9 New R Functions

`wilcox.test()` - performs the wilcox rank sum test on two datasets

```
starved <- c(1.9, 2.1, 3.8, 9.0, 9.6, 13.0, 14.7, 17.9,
            21.7, 29.0, 72.3)
fed <- c(1.5, 1.7, 2.4, 3.6, 5.7, 22.6, 22.8, 39.0, 54.4,
        72.1, 73.6, 79.5, 88.9)
```

```
wilcox.test(starved, fed, alternative = "less")
```

## Notes 11 New R Functions

`aov()` - fit an ANOVA model on two vectors of data

```
coag_time <- c(62, 60, 63, 59,
              63, 67, 71, 64, 65, 66,
              68, 66, 71, 67, 68, 68,
              56, 62, 60, 61, 63, 64, 63, 59)
poison <- c(rep("trt1", 4), rep("trt2", 6),
           rep("trt3", 6), rep("trt4", 8))
```

```
# Run ANOVA model and save the model object
poison_mod <- aov(coag_time ~ poison)
poison_mod
```

`summary()` - view basic summary output of an R model object

```
summary(poison_mod)
```

`fitted()` - extract vector of model fitted values `resid()` - extract vector of model residuals

```
fitted(poison_mod)
resid(poison_mod)

plot(fitted(poison_mod), resid(poison_mod),
     main = "Rat Poison Residuals vs Fitted")

qqnorm(resid(poison_mod),
       main = "Rat Poison Residual QQ-Plot")
```

qtukey() - find a specific quantile of the studentized range distribution for a Tukey post-hoc analysis

```
qtukey(0.95, nmeans = 4, df = 20)/sqrt(2)
```

kruskal.test() - perform the Kruskal-Wallis test on two vectors of data

```
so2 <- c(438, 619, 732, 638,
        857, 1014, 1153, 883, 1053,
        925, 786, 1179, 786,
        893, 891, 917, 695, 675, 595)
plants <- c(rep("plant1", 4), rep("plant2", 5),
           rep("plant3", 4), rep("plant4", 6))

kruskal.test(so2 ~ plants)
```

## Notes 12 New R Functions

cor() - find the correlation between two vectors of data

```
fathers <- c(71.3, 65.5, 65.9, 68.6, 71.4, 68.4, 65.0, 66.3,
            68.0, 67.3, 67.0, 69.3, 70.1, 66.9)
sons <- c(68.9, 67.5, 65.4, 68.2, 71.5, 67.6, 65.0, 67.0,
         65.3, 65.5, 69.8, 70.9, 68.9, 70.2)

cor(fathers, sons)
```

lm() - fit a linear model (regression line) on (x, y) data

```
height_mod <- lm(sons ~ fathers)
height_mod

# View more detailed output
summary(height_mod)

# Make residual plots
plot(fitted(height_mod), resid(height_mod))
qqnorm(resid(height_mod))
```

predict() - predict a y value based on a model

```
# By default, predict will return fitted values
predict(height_mod)

# Specify new data by including it in a dataframe object
new_fathers <- data.frame(fathers = 70)

predict(height_mod, newdata = new_fathers)
```

```

# Create 95% CI and PI by setting the "interval" argument
predict(height_mod, newdata = new_fathers,
         interval = "confidence")
predict(height_mod, newdata = new_fathers,
         interval = "prediction")

# We can also predict on multiple new values at once
new_fathers <- data.frame(fathers = c(66, 68, 70))

predict(height_mod, newdata = new_fathers,
         interval = "prediction")

```

## Notes 13 New R Functions

`pchisq()` - calculate a probability on the chi-squared distribution

```
pchisq(2.29, df = 4, lower.tail = F)
```

`chisq.test()` - use R to perform a chi-squared test

```

# Observed counts
obs <- c(85, 58, 61, 60, 55)

# Keep expected counts as probabilities
exp <- c(0.25, 0.2, 0.2, 0.2, 0.15)

# Use chisq.test to perform test automatically
chisq.test(obs, p = exp, correct = FALSE)

```

`matrix()` - build a matrix (table) of data with a specific number of rows and columns

```

# Build a matrix to hold categorical data
survey <- matrix(c(345, 135, 20, 222, 20, 8),
                 nrow = 2, ncol = 3, byrow = T)

chisq.test(survey, correct = F)

```