Lab Assignment: Chapter 18 - Basic Statistics

Michael Beebe

library(ggplot2)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(MASS)  
library(reshape2)  
library(plyr)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## The following object is masked from 'package:MASS':  
##   
## select

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## 1. Summary Stastics

Create a variable - stats - and store a random sample of 100 numbers between 100 and 200. Use replace=TRUE

stats <- sample(x = 100:200, size = 100, replace = TRUE)  
head(stats)

## [1] 139 141 137 118 115 156

Calculate the mean

mean(stats)

## [1] 151.5

Calculate the variance (2 ways)

sum((stats - mean(stats))^2) / (length(stats) - 1)

## [1] 695.2222

var(stats)

## [1] 695.2222

Calculate the standard deviation (2 ways)

sqrt(var(stats))

## [1] 26.36707

sd(stats)

## [1] 26.36707

Display the summary statistics

summary(stats)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 100.0 130.0 151.0 151.5 172.5 200.0

Display the 1st and 3rd quartiles in a different way than the summary statistics

quantile(x = stats, probs = c(.25, .75))

## 25% 75%   
## 130.0 172.5

A student’s grades are the following: quiz 1= 78, quiz 2 = 74, quiz 3 = 78, quiz 4 = 76, midterm = 68, and final = 90. The quizzes are weighted at 1/8 of the final grade and the exams are weighted at 1/4.

grades <- c(78, 74, 78, 76, 68, 90)  
weights <- c(1/8, 1/8, 1/8, 1/8, 1/4, 1/4)

* Display the mean.

mean(grades)

## [1] 77.33333

Display the weighted mean.

weighted.mean(x = grades, w = weights)

## [1] 77.75

## 2. Correlation and Covariance

Using the Old Faithful data, store the waiting time in a variable and the eruption duration in another

wait <- faithful$waiting  
duration <- faithful$eruptions

Find the correlation coefficient of eruption duration and waiting time in the data set faithful.

cor(wait, duration)

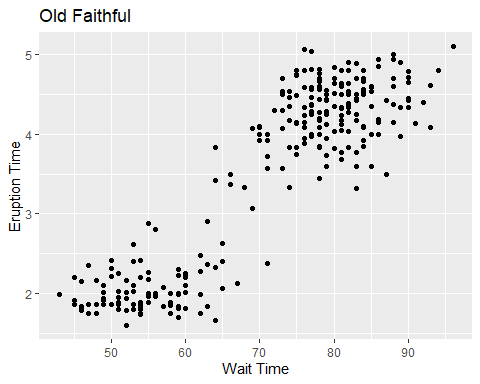
## [1] 0.9008112

cor(faithful)

## eruptions waiting  
## eruptions 1.0000000 0.9008112  
## waiting 0.9008112 1.0000000

Observe if there is any linear relationship between the variables.

ggplot(data.frame(x=wait, y=duration), aes(x=x, y=y)) +  
 geom\_point() +  
 labs(x ="Wait Time", y = "Eruption Time", title = "Old Faithful")



* We can see that there is very much so a linear correlation between wait time between eruptions and the duration of the eruption.

Using the same variables find the covariance, and analyze the results

cov(wait, duration)

## [1] 13.97781

cov(faithful)

## eruptions waiting  
## eruptions 1.302728 13.97781  
## waiting 13.977808 184.82331

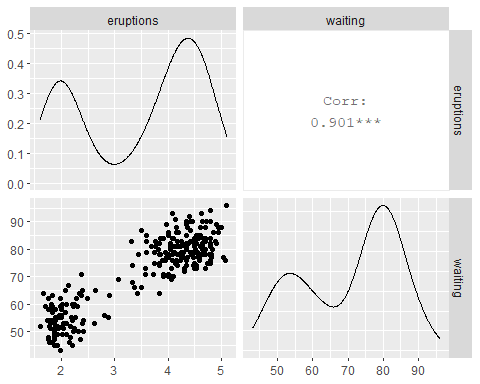
Use the cor function on a matrix to evaluate the faithful data.

mat <- cbind(wait, duration)  
cor(mat)

## wait duration  
## wait 1.0000000 0.9008112  
## duration 0.9008112 1.0000000

Graph the Old Faithful data

ggpairs(faithful)



## 3. T-Tests

Use the built-in data set named immer, which contains data on the barley yield in years 1931 and 1932 of a field.

head(immer)

## Loc Var Y1 Y2  
## 1 UF M 81.0 80.7  
## 2 UF S 105.4 82.3  
## 3 UF V 119.7 80.4  
## 4 UF T 109.7 87.2  
## 5 UF P 98.3 84.2  
## 6 W M 146.6 100.4

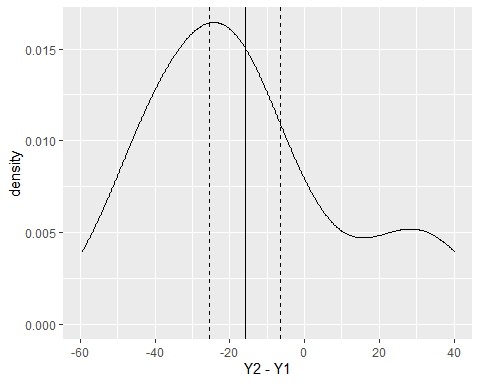
The data follows the normal distribution. Find the 95% confidence interval estimate of the difference between the mean barley yields.

t.test(immer$Y1, immer$Y2, paired = TRUE, conf.level = 0.95)

##   
## Paired t-test  
##   
## data: immer$Y1 and immer$Y2  
## t = 3.324, df = 29, p-value = 0.002413  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 6.121954 25.704713  
## sample estimates:  
## mean of the differences   
## 15.91333

Analyze the results

yieldDiff <- immer$Y2 - immer$Y1  
  
ggplot(immer, aes(x=Y2 - Y1)) +  
 geom\_density() +  
 geom\_vline(xintercept = mean(yieldDiff)) +  
 geom\_vline(xintercept = mean(yieldDiff) + 2\*c(-1,1) \* sd(yieldDiff) / sqrt(nrow(immer)), linetype=2)



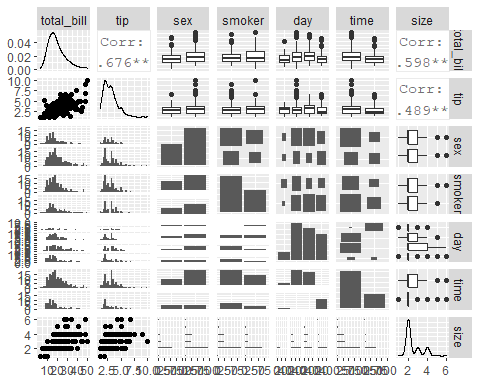
## 4. ANOVA

Complete the Analysis of Variance for the TIPS data

data(tips, package = "reshape2")  
head(tips)

## total\_bill tip sex smoker day time size  
## 1 16.99 1.01 Female No Sun Dinner 2  
## 2 10.34 1.66 Male No Sun Dinner 3  
## 3 21.01 3.50 Male No Sun Dinner 3  
## 4 23.68 3.31 Male No Sun Dinner 2  
## 5 24.59 3.61 Female No Sun Dinner 4  
## 6 25.29 4.71 Male No Sun Dinner 4

ggpairs(tips)



tipAnova <- aov(tip ~ day - 1, tips)  
tipIntercept <- aov(tip ~ day, tips)

tipAnova$coefficients

## dayFri daySat daySun dayThur   
## 2.734737 2.993103 3.255132 2.771452

tipIntercept$coefficients

## (Intercept) daySat daySun dayThur   
## 2.73473684 0.25836661 0.52039474 0.03671477

summary(tipAnova)

## Df Sum Sq Mean Sq F value Pr(>F)   
## day 4 2203.0 550.8 290.1 <2e-16 \*\*\*  
## Residuals 240 455.7 1.9   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

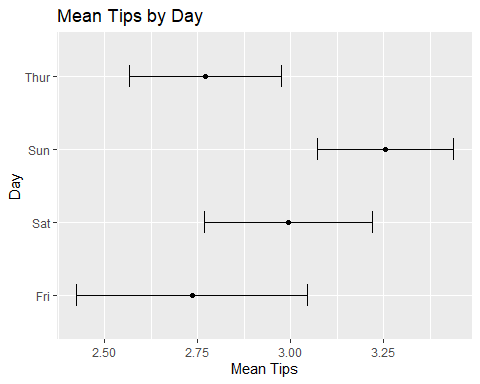
Analyze the p-values for each set of summary data.

tipsByDay <- ddply(  
 tips,  
 "day",  
 summarize,  
 tip.mean=mean(tip),  
 Length=NROW(tip),  
 tip.sd=sd(tip),  
 tfrac=qt(p=.90, df=Length-1),  
 Lower=tip.mean - tfrac\*tip.sd/sqrt(Length),  
 Upper=tip.mean + tfrac\*tip.sd/sqrt(Length)  
)  
tipsByDay

## day tip.mean Length tip.sd tfrac Lower Upper  
## 1 Fri 2.734737 19 1.019577 1.330391 2.423549 3.045925  
## 2 Sat 2.993103 87 1.631014 1.291473 2.767272 3.218934  
## 3 Sun 3.255132 76 1.234880 1.292941 3.071986 3.438277  
## 4 Thur 2.771452 62 1.240223 1.295585 2.567386 2.975517

Plot the dataset.

ggplot(tipsByDay, aes(x=tip.mean, y=day)) +  
 geom\_point() +  
 geom\_errorbarh(aes(xmin=Lower, xmax=Upper), height=.3) +  
 labs(x="Mean Tips", y="Day", title = {"Mean Tips by Day"})



Complete the Analysis of Variance for the Old Faithful data.

head(faithful)

## eruptions waiting  
## 1 3.600 79  
## 2 1.800 54  
## 3 3.333 74  
## 4 2.283 62  
## 5 4.533 85  
## 6 2.883 55

faithfulAnova <- aov(eruptions ~ waiting -1, faithful)  
faithfulIntercept <- aov(eruptions ~ waiting, faithful)

faithfulAnova$coefficients

## waiting   
## 0.05012919

faithfulIntercept$coefficients

## (Intercept) waiting   
## -1.87401599 0.07562795

summary(faithfulAnova)

## Df Sum Sq Mean Sq F value Pr(>F)   
## waiting 1 3561 3561 9621 <2e-16 \*\*\*  
## Residuals 271 100 0   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Analyze the p-values for each set of summary data.

eruptionsByWait <- ddply(  
 faithful,  
 "waiting",  
 summarize,  
 eruptions.mean=mean(eruptions),  
 Length=NROW(eruptions),  
 eruptions.sd=sd(eruptions),  
 tfrac=qt(p=.90, df=Length-1),  
 Lower=eruptions.mean - tfrac\*eruptions.sd/sqrt(Length),  
 Upper=eruptions.mean + tfrac\*eruptions.sd/sqrt(Length)  
)  
head(eruptionsByWait)

## waiting eruptions.mean Length eruptions.sd tfrac Lower Upper  
## 1 43 1.983000 1 NA NaN NaN NaN  
## 2 45 1.994667 3 0.17957264 1.885618 1.799173 2.190161  
## 3 46 1.883200 5 0.15053637 1.533206 1.779982 1.986418  
## 4 47 1.929250 4 0.28587104 1.637744 1.695158 2.163342  
## 5 48 1.928000 3 0.21508835 1.885618 1.693841 2.162159  
## 6 49 1.966800 5 0.09198478 1.533206 1.903729 2.029871

Plot the dataset.

ggpairs(faithful)

