Code\_for\_Prac\_1.R

u4321232

Fri Mar 23 09:16:58 2018

#Code for Prac 1  
  
#We will use following libraries (software packages written by someone else) for our purpose. It is usually a good practice to have the libraries set at the top of the script so that you and anyone else using the script can know upfron what software packages (libraries) will be used.  
######NOTE: If you haven't installed libraries listed below then you will need to install them first as follows. You will need internet connection for the installation and you will have to remove the comment character "#" from the front of the line.  
#install.packages("ggplot2")  
#install.packages("emmeans")  
  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.4.2

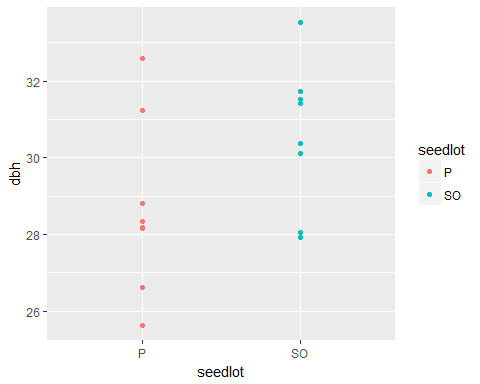
library(emmeans)

## Warning: package 'emmeans' was built under R version 3.4.3

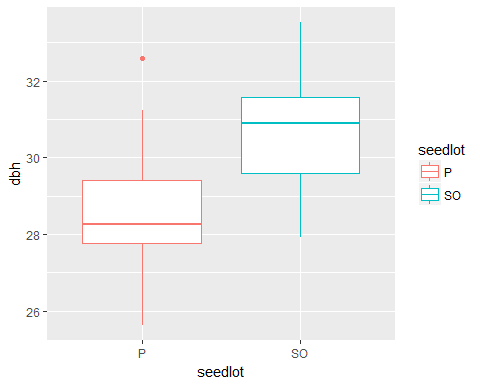
##We will set up a working directory as follows. The following directory path should be set to the directory path relevent to your computer.  
setwd("C:/Users/u4321232/Dropbox/RSB short course/Data")  
  
#Question #1: Seed orchard data  
  
#Import data using read.csv function. There are numerous functions and packages that come preinstalled with R and read.csv is one of them. On the bottom right corner of your RStudiom there is help tab. you can search for help regarding the usage of packages, libraries, function. Please try read.csv as a search term in the help tab and have a look at the variety of options available when reading a csv file  
data1 <- read.csv("Prac 1 seed orchard data.csv")  
  
#Check data structure using 'str'ucture command. One can have data.frame, matrix, vector, list and many more data structures that various functionalities in R can understand and manipulate. For now, data.frame should suffice.  
str(data1)

## 'data.frame': 16 obs. of 3 variables:  
## $ plot : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ seedlot: Factor w/ 2 levels "P","SO": 2 2 2 2 2 2 2 2 1 1 ...  
## $ dbh : num 30.4 27.9 28.1 31.4 30.1 ...

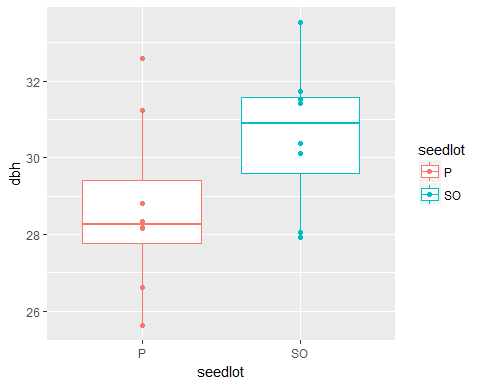
#Visualise data using ggplot  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+geom\_point()



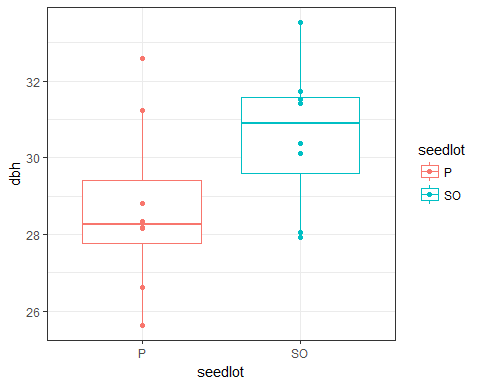
#you can create box plots  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+geom\_boxplot()



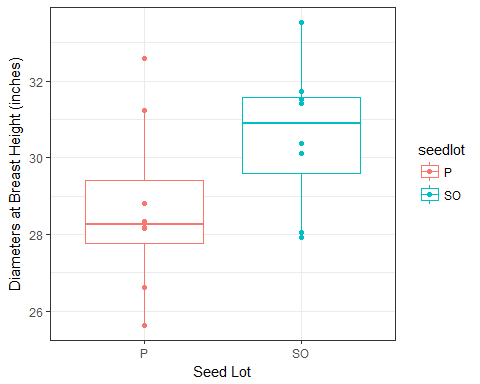
#you can combine boxplot and point plot  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+geom\_boxplot()+geom\_point()



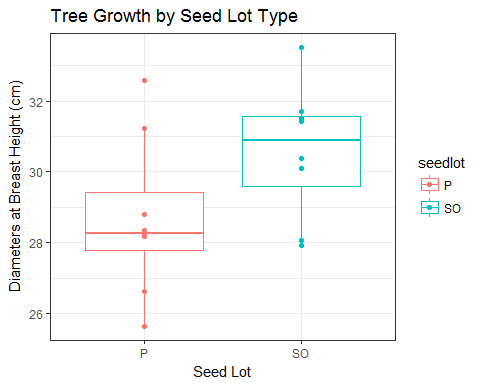
#you can change the background to black and white  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+geom\_boxplot()+geom\_point()+theme\_bw()



#you can modify axis titles  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+geom\_boxplot()+geom\_point()+theme\_bw()+xlab("Seed Lot")+ylab("Diameters at Breast Height (inches)")



#you can add title to the plot  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+  
 geom\_boxplot()+geom\_point()+theme\_bw()+xlab("Seed Lot")+  
 ylab("Diameters at Breast Height (cm)")+labs(title="Tree Growth by Seed Lot Type")



#If you are happy with the plot and want to export the plot then you can do so by using the Export tab in the bottom right corner panel where plots are usually displayed.  
#Programmatically, you can do so to export a PDF file  
#open a pdf document with height 4 and width 6 inches.  
pdf("seedlot.boxplot.pdf", height=4,width=6)  
#draw the plot to the PDF file  
ggplot(data1,aes(seedlot,dbh,colour=seedlot))+geom\_boxplot()+geom\_point()+  
 theme\_bw()+xlab("Seed Lot")+ylab("Diameter at Breast Height (inches)")+  
 labs(title="Tree Growth by Seed Lot Type")  
##close the PDF file. Dont forget to do this as it will keep writing to the same file.  
dev.off()

## png   
## 2

#you can check out the newly created PDF file in the directory that was set intially at the top  
##more information about the various ggplot functions and modification, go to http://docs.ggplot2.org/current/  
##I would usually just google for a particular question such as "how to modify legend title ggplot2"  
##There are more than many ways to achieve the same task with ggplot2, so feel free to select the answer   
##that suits you best. There is no right or wrong way of doing it.  
  
  
#Fit model, inference, and estimate means (SE)  
model1<-lm(dbh~seedlot, data=data1)  
anova(model1)

## Analysis of Variance Table  
##   
## Response: dbh  
## Df Sum Sq Mean Sq F value Pr(>F)   
## seedlot 1 14.27 14.2695 3.2531 0.09284 .  
## Residuals 14 61.41 4.3864   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(model1)

##   
## Call:  
## lm(formula = dbh ~ seedlot, data = data1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.0725 -0.9200 -0.2769 0.9887 3.8975   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 28.6925 0.7405 38.749 1.21e-15 \*\*\*  
## seedlotSO 1.8888 1.0472 1.804 0.0928 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.094 on 14 degrees of freedom  
## Multiple R-squared: 0.1886, Adjusted R-squared: 0.1306   
## F-statistic: 3.253 on 1 and 14 DF, p-value: 0.09284

emmeans(model1,~seedlot)

## seedlot emmean SE df lower.CL upper.CL  
## P 28.69250 0.7404743 14 27.10434 30.28066  
## SO 30.58125 0.7404743 14 28.99309 32.16941  
##   
## Confidence level used: 0.95

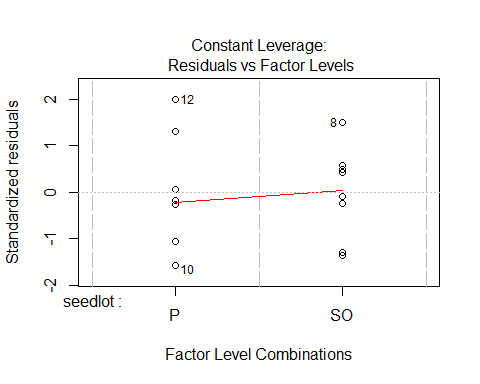
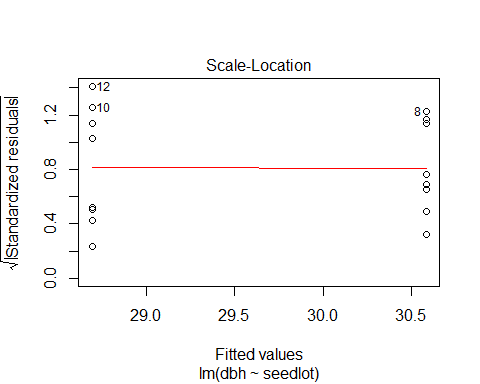
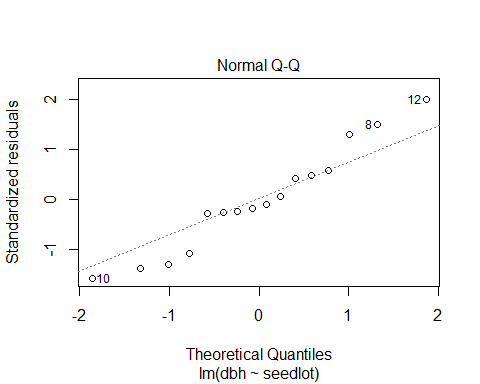
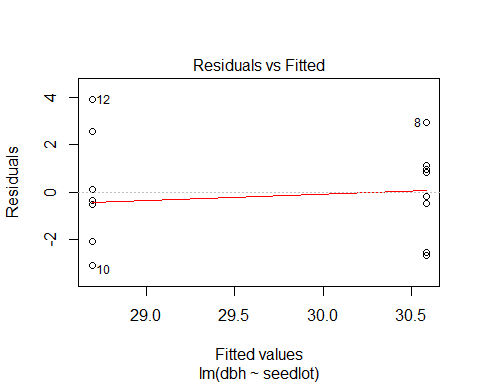
##what is the data 'str'ucture for model1  
str(model1)

## List of 13  
## $ coefficients : Named num [1:2] 28.69 1.89  
## ..- attr(\*, "names")= chr [1:2] "(Intercept)" "seedlotSO"  
## $ residuals : Named num [1:16] -0.201 -2.671 -2.521 0.839 -0.471 ...  
## ..- attr(\*, "names")= chr [1:16] "1" "2" "3" "4" ...  
## $ effects : Named num [1:16] -118.5475 -3.7775 -2.0425 1.3175 0.0075 ...  
## ..- attr(\*, "names")= chr [1:16] "(Intercept)" "seedlotSO" "" "" ...  
## $ rank : int 2  
## $ fitted.values: Named num [1:16] 30.6 30.6 30.6 30.6 30.6 ...  
## ..- attr(\*, "names")= chr [1:16] "1" "2" "3" "4" ...  
## $ assign : int [1:2] 0 1  
## $ qr :List of 5  
## ..$ qr : num [1:16, 1:2] -4 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 0.25 ...  
## .. ..- attr(\*, "dimnames")=List of 2  
## .. .. ..$ : chr [1:16] "1" "2" "3" "4" ...  
## .. .. ..$ : chr [1:2] "(Intercept)" "seedlotSO"  
## .. ..- attr(\*, "assign")= int [1:2] 0 1  
## .. ..- attr(\*, "contrasts")=List of 1  
## .. .. ..$ seedlot: chr "contr.treatment"  
## ..$ qraux: num [1:2] 1.25 1.2  
## ..$ pivot: int [1:2] 1 2  
## ..$ tol : num 1e-07  
## ..$ rank : int 2  
## ..- attr(\*, "class")= chr "qr"  
## $ df.residual : int 14  
## $ contrasts :List of 1  
## ..$ seedlot: chr "contr.treatment"  
## $ xlevels :List of 1  
## ..$ seedlot: chr [1:2] "P" "SO"  
## $ call : language lm(formula = dbh ~ seedlot, data = data1)  
## $ terms :Classes 'terms', 'formula' language dbh ~ seedlot  
## .. ..- attr(\*, "variables")= language list(dbh, seedlot)  
## .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. ..$ : chr [1:2] "dbh" "seedlot"  
## .. .. .. ..$ : chr "seedlot"  
## .. ..- attr(\*, "term.labels")= chr "seedlot"  
## .. ..- attr(\*, "order")= int 1  
## .. ..- attr(\*, "intercept")= int 1  
## .. ..- attr(\*, "response")= int 1  
## .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. ..- attr(\*, "predvars")= language list(dbh, seedlot)  
## .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "factor"  
## .. .. ..- attr(\*, "names")= chr [1:2] "dbh" "seedlot"  
## $ model :'data.frame': 16 obs. of 2 variables:  
## ..$ dbh : num [1:16] 30.4 27.9 28.1 31.4 30.1 ...  
## ..$ seedlot: Factor w/ 2 levels "P","SO": 2 2 2 2 2 2 2 2 1 1 ...  
## ..- attr(\*, "terms")=Classes 'terms', 'formula' language dbh ~ seedlot  
## .. .. ..- attr(\*, "variables")= language list(dbh, seedlot)  
## .. .. ..- attr(\*, "factors")= int [1:2, 1] 0 1  
## .. .. .. ..- attr(\*, "dimnames")=List of 2  
## .. .. .. .. ..$ : chr [1:2] "dbh" "seedlot"  
## .. .. .. .. ..$ : chr "seedlot"  
## .. .. ..- attr(\*, "term.labels")= chr "seedlot"  
## .. .. ..- attr(\*, "order")= int 1  
## .. .. ..- attr(\*, "intercept")= int 1  
## .. .. ..- attr(\*, "response")= int 1  
## .. .. ..- attr(\*, ".Environment")=<environment: R\_GlobalEnv>   
## .. .. ..- attr(\*, "predvars")= language list(dbh, seedlot)  
## .. .. ..- attr(\*, "dataClasses")= Named chr [1:2] "numeric" "factor"  
## .. .. .. ..- attr(\*, "names")= chr [1:2] "dbh" "seedlot"  
## - attr(\*, "class")= chr "lm"

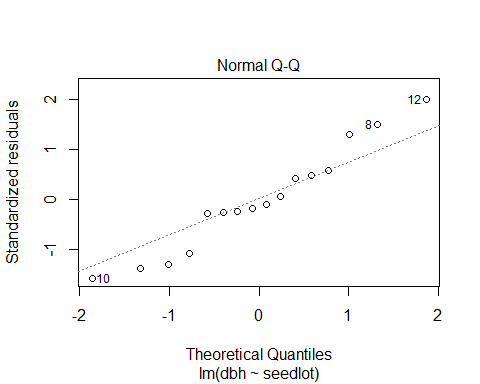
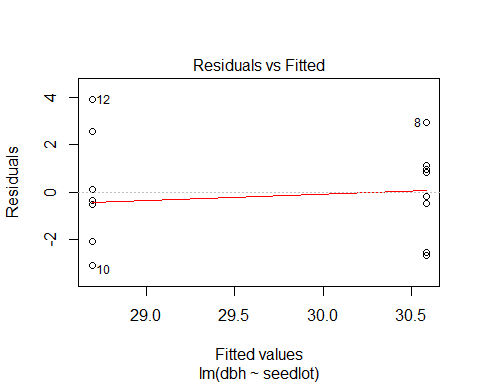
##it shows as List of 13  
##what are the items in list  
names(model1)

## [1] "coefficients" "residuals" "effects" "rank"   
## [5] "fitted.values" "assign" "qr" "df.residual"   
## [9] "contrasts" "xlevels" "call" "terms"   
## [13] "model"

#Assess model assumptions  
plot(model1)



##NOTE: multiple plots are drawn when you hit enter. Once you are familiar with the type of plots and information content of plots, you can choose which plot to draw. For example we know that we would want the first plot only then we can use the following  
plot(model1, which=1:2)



##again you can export the plot into PDF if you wish as follows  
pdf("fitted.vs.residuals.pdf")  
plot(model1, which=1)  
dev.off()

## png   
## 2

##you can plot various aspects of the fitted model using ggplot2 for exploration by following help at http://docs.ggplot2.org/0.9.3.1/fortify.lm.html  
##first we modify the data in the "model1" object for syntax that is acceptable by ggplot2  
ggmodel1 <- fortify(model1,data1)  
ggplot(ggmodel1, aes(.fitted,.resid))+geom\_point()+geom\_hline(yintercept=0)+geom\_smooth(se=FALSE)

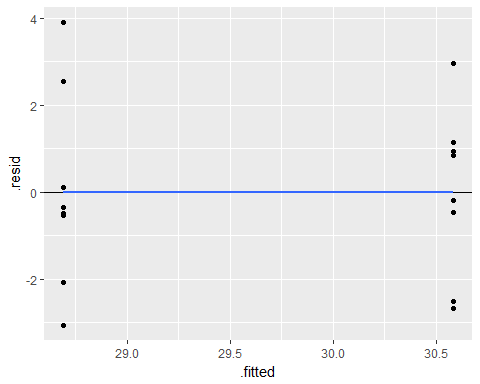
## `geom\_smooth()` using method = 'loess'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : pseudoinverse used at 28.683

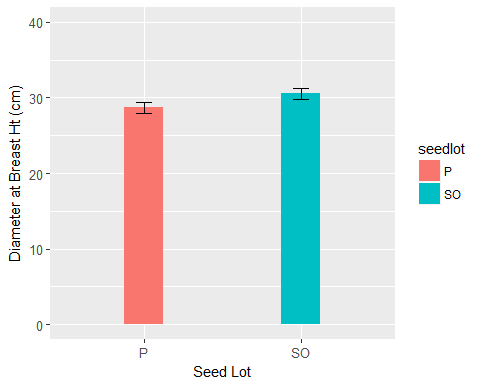
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : neighborhood radius 1.8982

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =  
## parametric, : There are other near singularities as well. 3.6031



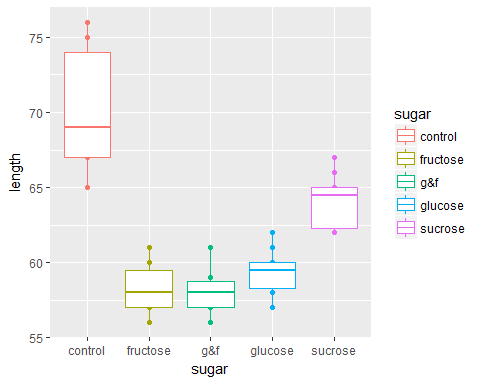
##get the emmeans output converted to a data.frame for easy understanding and manipulation  
summarisedinfo <- summary(emmeans(model1,~seedlot))  
ggplot(summarisedinfo,aes(seedlot,emmean,fill=seedlot))+geom\_bar(stat="identity", width=.25)+  
 ylim(0,40)+geom\_errorbar(aes(ymin=emmean-SE,ymax=emmean+SE),width=0.10) +   
 theme(axis.text=element\_text(size=10)) + xlab("Seed Lot") + ylab("Diameter at Breast Ht (cm)")



###once the plots are obtained, you can also write the tabular output to a file  
write.table(summarisedinfo, file="summary.txt",sep="\t", row.names=F, quote = F)  
  
##you can check help pages for any function by putting a question mark before the name of the function.  
#?write.table  
  
  
#Question #2 Pea data  
#Import data and check data types  
data2 <- read.csv("Prac 1 pea data.csv")  
str(data2)

## 'data.frame': 50 obs. of 3 variables:  
## $ sampleNo: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ sugar : Factor w/ 5 levels "control","fructose",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ length : int 75 67 70 75 65 71 67 67 76 68 ...

#Visualise data  
ggplot(data2,aes(sugar,length,colour=sugar))+geom\_point()+geom\_boxplot()



#Fit model, inference, and estimate means (SE)  
model2<-lm(length~sugar, data=data2)  
anova(model2)

## Analysis of Variance Table  
##   
## Response: length  
## Df Sum Sq Mean Sq F value Pr(>F)   
## sugar 4 1077.3 269.330 49.368 6.737e-16 \*\*\*  
## Residuals 45 245.5 5.456   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

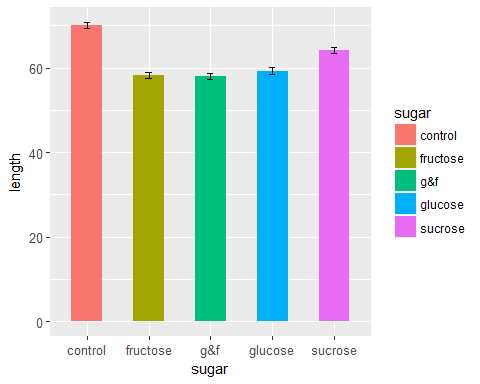
summary(model2)

##   
## Call:  
## lm(formula = length ~ sugar, data = data2)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -5.100 -1.825 -0.150 0.975 5.900   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 70.1000 0.7386 94.907 < 2e-16 \*\*\*  
## sugarfructose -11.9000 1.0446 -11.392 7.50e-15 \*\*\*  
## sugarg&f -12.1000 1.0446 -11.584 4.27e-15 \*\*\*  
## sugarglucose -10.8000 1.0446 -10.339 1.81e-13 \*\*\*  
## sugarsucrose -6.0000 1.0446 -5.744 7.48e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.336 on 45 degrees of freedom  
## Multiple R-squared: 0.8144, Adjusted R-squared: 0.7979   
## F-statistic: 49.37 on 4 and 45 DF, p-value: 6.737e-16

emmeans(model2,pairwise~sugar)

## $emmeans  
## sugar emmean SE df lower.CL upper.CL  
## control 70.1 0.7386173 45 68.61235 71.58765  
## fructose 58.2 0.7386173 45 56.71235 59.68765  
## g&f 58.0 0.7386173 45 56.51235 59.48765  
## glucose 59.3 0.7386173 45 57.81235 60.78765  
## sucrose 64.1 0.7386173 45 62.61235 65.58765  
##   
## Confidence level used: 0.95   
##   
## $contrasts  
## contrast estimate SE df t.ratio p.value  
## control - fructose 11.9 1.044563 45 11.392 <.0001  
## control - g&f 12.1 1.044563 45 11.584 <.0001  
## control - glucose 10.8 1.044563 45 10.339 <.0001  
## control - sucrose 6.0 1.044563 45 5.744 <.0001  
## fructose - g&f 0.2 1.044563 45 0.191 0.9997  
## fructose - glucose -1.1 1.044563 45 -1.053 0.8291  
## fructose - sucrose -5.9 1.044563 45 -5.648 <.0001  
## g&f - glucose -1.3 1.044563 45 -1.245 0.7256  
## g&f - sucrose -6.1 1.044563 45 -5.840 <.0001  
## glucose - sucrose -4.8 1.044563 45 -4.595 0.0003  
##   
## P value adjustment: tukey method for comparing a family of 5 estimates

#Set up graph summarising the model  
sum\_model <- summary(emmeans(model2,~sugar))  
ggplot(sum\_model,aes(sugar,emmean,fill=sugar))+geom\_bar(stat="identity", width=.5)+  
 geom\_errorbar(aes(ymin=emmean-SE,ymax=emmean+SE),width=0.10) +   
 theme(axis.text=element\_text(size=10)) + xlab("sugar") +   
 ylab("length")



#Assess model assumptions  
plot(model2,1)

