Charts

#Loading packages:tidyverse and ggplot2 are neccesary for the creation of figures, and finalfit is used to help download this RMarkdown file as a word document. extrafont and the font\_import command were used to customize the fonts on our figures to match that used in the poster# #For future runs of the data the font steps are unneccesary. When you run this chunk in R this prompt will appear in your console: ‘Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system. Continue? [y/n]’ Type ‘n’ for no and R will skip these commands#

rm(list=ls())  
library("tidyverse")

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.1 ──

## ✓ ggplot2 3.3.5 ✓ purrr 0.3.4  
## ✓ tibble 3.1.3 ✓ dplyr 1.0.7  
## ✓ tidyr 1.1.3 ✓ stringr 1.4.0  
## ✓ readr 2.0.0 ✓ forcats 0.5.1

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library("ggplot2")  
library("extrafont")

## Registering fonts with R

font\_import()

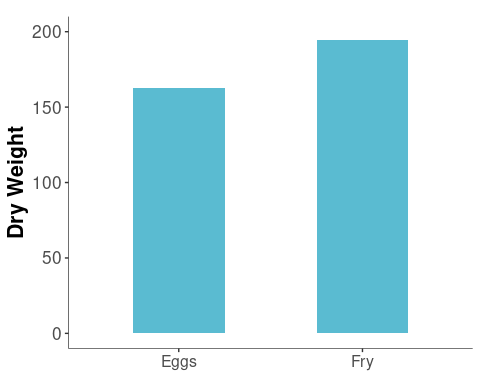
## Importing fonts may take a few minutes, depending on the number of fonts and the speed of the system.  
## Continue? [y/n]

## Exiting.

library("finalfit")

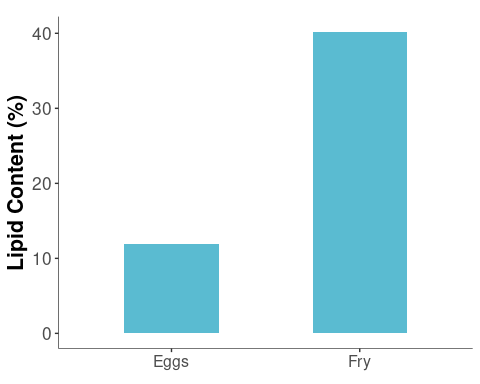
#DRY WEIGHT: Bar Chart# #The data used in this CSV file was the average dry weight per unit of three egg samples (F1C, F4C, F01) and the average dry weight for three fry samples (M3C, M01, M13). This excluded outliers M1C and F06#

## Warning in read.table(file = file, header = header, sep = sep, quote = quote, :  
## incomplete final line found by readTableHeader on 'For Bar Charts - Dry Weight  
## Means.csv'



#LIPIDS: Mass# #The data used in this CSV file was the mass of the lipid content for three egg samples (F1C, F4C, F01) and the mass of the lipid content for three fry samples (M1C, M01, M13). This excluded outliers M3C and F06#

lipid.content <- read.csv("For Bar Charts - Lipid Content Mass.csv", stringsAsFactors = T)  
  
ggplot(lipid.content  
 , aes(x=Type, y=Mass)) +  
 geom\_col(size = 3, width = 0.5, aes(fill="Type")) +  
 scale\_fill\_manual(values = c("Type" = "#5abbd1")) +  
 ylab("Lipid Content (%)")+ggtitle("Percent of Lipid Content of the Total Mass")+  
 theme(axis.title.x = element\_text(size=0),   
 axis.title.y = element\_text(hjust = 0.5, size=16, face = "bold"),   
 axis.text.y = element\_text(size=13),  
 axis.text.x = element\_text(size=12),   
 legend.text = element\_text(size=0),  
 legend.title = element\_text(size=0),  
 legend.position = "none",  
 plot.title = element\_text(hjust = 0.5, size=0),  
 axis.line.x.bottom = element\_line(size=0.2),  
 axis.line.y.left = element\_line(size=0.2),  
 panel.background = element\_rect(fill="white"),  
 plot.background = element\_rect(fill="white"),  
 legend.background = element\_rect(fill="white"))



#Lipids: Standard Curve# #This is the calibration curve for the lipid standards. The data used for this figure was from the first round of standards we ran on 8/3/21 (the second round was run on 8/5/21). In this chunck of code I also ran a linear regression (line 82), which proved to be unneccesary because ggplot2 can perform the same command#

lipid.standard <- read.csv("For Bar Charts - Lipid Standards.csv", stringsAsFactors = T)  
  
lipid.standard.lm <- lm(Concentration ~ Average.Abs., lipid.standard)  
  
ggplot(lipid.standard  
 , aes(x=Concentration, y=Average.Abs.)) +  
 geom\_point(size = 2, color = "#5abbd1") +  
 geom\_smooth(method="lm", color = "#004959", se = FALSE, linetype = "dashed", size = 0.75) +  
 xlab("Concentration (%)")+ylab("Absorbance (A.U.)")+ggtitle("Lipid Standard Curve")+  
 xlim(0,1.1)+ylim(0,0.2)+  
 theme(axis.title.x = element\_text(size=16, face = "bold"),   
 axis.title.y = element\_text(size=16, face = "bold"),   
 axis.text.y = element\_text(size=12),  
 axis.text.x = element\_text(size=12),   
 legend.text = element\_text(size=13),  
 legend.title = element\_text(size=18),  
 plot.title = element\_text(hjust = 0.5, size=0),  
 axis.line.x.bottom = element\_line(size=0.2),  
 axis.line.y.left = element\_line(size=0.2),  
 panel.background = element\_rect(fill="white"),  
 legend.background = element\_rect(fill="white"))

## `geom\_smooth()` using formula 'y ~ x'

