couldn’t figure out how to get RMarkdown to work on my computer, so I copied and pasted my script into Word instead…

```
# load libraries

library(tidyverse)

library(readr)

library(lubridate)

library(magrittr)

library(ggplot2)

library(car)

library(emmeans)

library(lme4)

library(ggthemes)

library(plyr) #HSR note- if you need plyr, need to load it before tidyverse. But dplyr in tidyverse does everything plyr can do and more, so shoudln't need plyr.


##read in data

prelim_tidy <- read_csv("_data/_tidy/prelim_data_tidycols.csv")

View(prelim_tidy)

nrow(prelim_tidy) #2873 rows


#Omit any missing values:

summary(is.na(prelim_tidy)) #looks like missing lots of therm data


prelim_temp<-na.omit(prelim_tidy)

View(prelim_temp)

nrow(prelim_temp) #810 rows; most rows were removed bc therm data missing.


prelim_temp$therm_t[prelim_temp$therm_t >= 42] <- NA #remove therm values >42 because likely a temp sensor problem? 6 rows removed.
```

```
prelim_temp<-na.omit(prelim_temp) #remove those rows with NA's

View(prelim_temp)

nrow(prelim_temp) #now 804 rows


#Combine positional behaviors into smaller categories

# suspensory = Cb, Br & QM

#Vertical climb and cling together

prelim_temp$pos_beh[prelim_temp$pos_beh == "Cb"] <- "Su"

prelim_temp$pos_beh[prelim_temp$pos_beh == "Br"] <- "Su"

prelim_temp$pos_beh[prelim_temp$pos_beh == "QM"] <- "Su"

prelim_temp$pos_beh[prelim_temp$pos_beh == "Vci"] <- "VC"

prelim_temp$pos_beh[prelim_temp$pos_beh == "Abp"] <- "Bp"

prelim_temp$pos_beh[prelim_temp$pos_beh == "BpS"] <- "Bp"

prelim_temp$pos_beh[prelim_temp$pos_beh == "BpW"] <- "Bp"



#############

##re-order levels

prelim_temp$time_od <- factor(prelim_temp$time_od, levels = c("e_morning", "l_morning",
"e_afternoon", "l_afternoon", "evening"))

prelim_temp$pos_beh <- factor(prelim_temp$pos_beh, levels = c("QS", "Ly", "Sq", "St", "QW", "Bp",
"Su", "VC"))


## Change to factor and numeric

factor_cols <- c("pos_beh","context", "substrate", "hab_type", "individual")

numeric_cols <- c("sun", "therm_t", "t_lo", "t_hi")


prelim_temp[factor_cols] <- lapply(prelim_temp[factor_cols], as.factor)
```

```
prelim_temp[numeric_cols] <- lapply(prelim_temp[numeric_cols], as.numeric)


str(prelim_temp)


##plotting

## Change levels for the plot

prelim_temp$pos_beh <- factor(prelim_temp$pos_beh, levels = c("Ly","St", "Sq", "QS", "QW", "Bp",
"Su", "VC"))


pb_raw_plot <- ggplot(prelim_temp, aes(pos_beh, therm_t)) +
        geom_boxplot(fill = "white", colour = "blue", size = 1) +
        ggtitle("Effect of Positional Behavior on Body Temperature (Raw)") +
        scale_x_discrete(name = "Positional Behavior", labels = c("Lie", "Sit",
                "Squat", "Quad. Stand", "Quad. Walk", "Bipedal",
                "Suspensory", "Vert. Climb/Cling")) +
        scale_y_continuous(name = "Body Temperature") +
        theme_classic()
pb_raw_plot


##save plot

ggsave("pb_raw_plot.pdf", width=10, height=6, units="in")


ggsave("pb_raw_plot.png", width=10, height=6, units="in")


#labels = c("Lie", "Sit", "Squat", "Quad. Stand", "Quad. Walk", "Bipedal", "Suspensory", "Vert.
Climb/Cling"


ggplot(prelim_temp, aes(individual, therm_t)) +
```

```
  geom_boxplot() +
  labs(x="Individual", y="Body Temperature") +
  theme_classic()

ggplot(prelim_temp, aes(hab_type, therm_t)) +
  geom_boxplot() +
  labs(x="Habitat Type", y="Body Temperature") +
  theme_classic()

ggplot(prelim_temp, aes(time_od, therm_t)) +
  geom_boxplot() +
  labs(x="Time of Day", y="Body Temperature") +
  theme_classic()

##counts
ggplot(prelim_temp, aes(pos_beh))+
  geom_bar(stat="count") +
  labs(x="Positional Behavior", y="Count") +
  theme_classic()

ggplot(prelim_temp, aes(hab_type))+
  geom_bar(stat="count") +
  labs(x="Habitat Type", y="Count")+
  theme_minimal()    ###can I combine bamboo with bamboo woodland?

ggplot(prelim_temp, aes(time_od))+
  geom_bar(stat="count") +
  labs(x="Time of Day", y="Count")+
  theme_minimal()
```

```
ggplot(prelim_temp, aes(individual))+
  geom_bar(stat="count") +
  labs(x="Individual", y="Count")+
  theme_minimal()

ggplot(prelim_temp, aes(sun))+
  geom_histogram()

###model plotting
library(broom)

m1 <- lmer(therm_t ~ pos_beh + time_od + sun + date + hab_type +
      (1|individual), data = prelim_temp)
summary(m1)
confint(m1)

#ggplot(fortify(m1), aes(pos_beh, therm_t)) +
  #stat_summary(fun.data=mean_se, geom="pointrange") +
  #stat_summary(aes(y=.fitted), fun.y=mean, geom="line")

#predict(m1, preddata)

#m1_res <- tidy(coef(summary(m1)) )
#ggplot(m1_res, aes(pos_beh, therm_t)) +
  #geom_point()

ggplot(prelim_temp, aes(pos_beh, therm_t)) +
  geom_point(size = 3) +
```

```
  geom_point(aes(y=predict(m1)), size = 1)


#change levels for the plot

prelim_temp$pos_beh <- factor(prelim_temp$pos_beh, levels = c("Ly","St", "Sq", "QS", "QW", "Bp",
"Su", "VC"))


pred_v_obs <- ggplot(prelim_temp, aes(pos_beh, therm_t)) +

        geom_point(size = 3, colour = "blue") + #shows raw data

        geom_boxplot(colour="blue")+ #shows raw data

        geom_boxplot(aes(y=predict(m1)), size = 1, colour = "red") + #shows model predictions

        scale_x_discrete(name = "Positional Behavior", labels = c("Lie", "Sit",

                        "Squat", "Quad. Stand", "Quad. Walk", "Bipedal",

                        "Suspensory", "Vert. Climb/Cling")) +

        scale_y_continuous(name = "Body Temperature") +

        theme_classic()


pred_v_obs

#hsr: what this tells me is that the model predictions are pretty close to the raw data. (red and blue
boxplots are similar). You can plot just the raw data and see the pattern, if you'd like.


##save plot for reference

ggsave("pred_v_obs.pdf", width=10, height=6, units="in")


ggsave("pred_v_obs.png", width=10, height=6, units="in")


####################

##change levels back

prelim_temp$pos_beh <- factor(prelim_temp$pos_beh, levels = c("QS", "Ly", "Sq", "St", "QW", "Bp",
"Su", "VC"))
```

```
##tried another thing and it didn't work...
#newdat <- expand.grid(pos_beh=unique(prelim_temp$pos_beh),
                #time_od=unique(prelim_temp$time_od),
                #sun=c(min(prelim_temp$sun),
                 #     max(prelim_temp$sun)),
                #date=date(prelim_temp$date),
                #hab_type=unique(prelim_temp$hab_type))

#ggplot(prelim_temp, aes(x=pos_beh, y=therm_t)) +
 # geom_point(size=3) +
  #geom_line(data=newdat, aes(y=predict(m1, level=0, newdata=newdat), size="Positional Behavior")) +
  #scale_size_manual(name="Predictions", values=c("Positional Behavior"=3)) +
  #theme_classic()

## Create dataframe to predict over
##attempting to follow tutorial but got too confused....
preddata <- with(prelim_temp, expand.grid(pos_beh = levels(pos_beh), time_od = levels(time_od), sun = c(0,50,100), date = c("2018-06-28", "2018-06-06", "2018-07-16"), hab_type = c("WD", "WG", "GL"), individual = c("DW","JM", "LT")))

factor_cols <- c("pos_beh", "time_od", "hab_type", "individual")
numeric_cols <- c("sun")

preddata[factor_cols] <- lapply(preddata[factor_cols], as.factor)
preddata[numeric_cols] <- lapply(preddata[numeric_cols], as.numeric)

preddata$date <- as.Date(preddata$date)

str(preddata)
```

```
########
mm <- model.matrix(~pos_beh + time_od + sun + date + hab_type, data=preddata)

###won't work because I don't have 2 or more levels per factor

pframe2 <- data.frame(preddata,eta=mm%*%fixef(m1))
pframe2 <- with(pframe2,data.frame(pframe2,prop=eta))
pvar1 <- diag(mm %*% tcrossprod(vcov(m1),mm))
tvar1 <- pvar1+VarCorr(m1)$individual
pframe2 <- data.frame(
  pframe2
  , plo = pframe2$eta-2*sqrt(pvar1)
  , phi = pframe2$eta+2*sqrt(pvar1)
  , tlo = pframe2$eta-2*sqrt(tvar1)
  , thi = pframe2$eta+2*sqrt(tvar1)
)
```