```
rm(list=ls())
library(GGally)
library(qqplotr)
library(corrplot)
library(car)
library(reshape)
library(tidyverse)
library(magrittr)
library(dplyr)
library(betareg)
library(statmod)
library(jtools)
library(numDeriv)
library(latex2exp)
library(broom.mixed)
library(gridExtra)
library(grid)
library(lattice)
library(ggplot2)
library(caret)
library(stringr)

if (Sys.getenv('USER') == "mortenjohnsen"){
  setwd("/Users/mortenjohnsen/OneDrive - Danmarks Tekniske Universitet/DTU/10.
Semester/02424 - Advanced Dataanalysis and Statistical Modellling/02424---
Assignments/Assignment 2/")
  figpath <- "/Users/mortenjohnsen/OneDrive - Danmarks Tekniske Universitet/DTU/10.
Semester/02424 - Advanced Dataanalysis and Statistical Modellling/02424---
Assignments/Assignment 2/figs/"
}else if (Sys.getenv('USER') == "freja"){
  setwd("~/Documents/Uni/TiendeSemester/Adv. data analysis and stat. modelling/02424---
Assignments/Assignment 2")
  figpath <- "~/Documents/Uni/TiendeSemester/Adv. data analysis and stat.
modelling/02424---Assignments/Assignment 2/figs/"
}else{
  setwd("C:/Users/catdu/OneDrive/DTU/10. semester/Advanced Dataanalysis and Statistical
Modelling/Assignment 1/02424---Assignments/Assignment 2/")
  figpath <- "C:/Users/catdu/OneDrive/DTU/10. semester/Advanced Dataanalysis and
Statistical Modelling/Assignment 1/02424---Assignments/Assignment 2/"
}

clothing <- read.csv(file = "clothing.csv", header = T)

#### Clothing ####
#1)
c.data <- dplyr::select(clothing, -subjId, -day, -time, -X)
head(c.data)

melt(c.data) %>%
  ggplot()+
  geom_histogram(aes(x = value, fill = sex), position = "identity", alpha = 0.4)+
  scale_fill_manual(values = c("blue", "orange"))+
  facet_wrap(~variable, scales = "free")+
  theme_bw()
ggsave(paste0(figpath,"clothing_data_histograms.png"), width = 30, height = 10, units =
"cm")

################################ 1) Distributions ################################
beta.dist <- function(theta){
  return(-sum(dbeta(c.data$clo, shape1 = theta[1], shape2 = theta[2], log = T)))
}

norm.dist <- function(theta){
  return(-sum(dnorm(c.data$clo, mean = theta[1], sd = theta[2], log = T)))
}

gamma.dist <- function(theta){
  return(-sum(dgamma(c.data$clo, shape = theta[1], rate = theta[2], log = T)))
}

invgaus.dist <- function(theta){
  return(-sum(dinvgauss(c.data$clo, mean = theta[1], dispersion = theta[2], log =T)))
}

lnorm.dist <- function(theta){
  return(-sum(dlnorm(x=c.data$clo, meanlog = theta[1], sdlog = theta[2], log = T)))
}

beta.hat <- nlminb(start = c(1,1), objective = beta.dist)
norm.hat <- nlminb(start = c(1,1), objective = norm.dist)
gamma.hat <- nlminb(start = c(1,1), objective = gamma.dist)
invgaus.hat <- nlminb(start = c(1,1), objective = invgaus.dist)
lnorm.hat <- nlminb(start = c(1,1), objective = lnorm.dist)

ggplot(c.data)+
  geom_histogram(aes(x = clo, y = after_stat(density)), colour = "white",
                 position = "identity", alpha = 0.4)+
  scale_fill_manual(values = c("blue", "orange"))+
  stat_function(aes(colour = "1: beta"), fun = dbeta,
                args = list(shape1 = beta.hat$par[1],
                            shape2 = beta.hat$par[2]))+
  stat_function(aes(colour = "2: normal"), fun = dnorm,
                args = list(mean = norm.hat$par[1],
                            sd = norm.hat$par[2]))+
  stat_function(aes(colour = "3: gamma"), fun = dgamma,
                args = list(shape = gamma.hat$par[1],
                            rate = gamma.hat$par[2]))+
  stat_function(aes(colour = "4: gamma"), fun = dinvgauss,
                args = list(mean = invgaus.hat$par[1],
                            dispersion = invgaus.hat$par[2]))+
  stat_function(aes(colour = "5: log-normal"), fun = dlnorm,
                args = list(meanlog = lnorm.hat$par[1],
                            sdlog = lnorm.hat$par[2]))+
  theme_bw()+
  labs(y = "", colour = "Distribution")+
  scale_colour_manual(
    values = c("blue", "orange", "black", "red", "purple", "grey"),
    labels = c(
      paste0("Beta [AIC: ",
             round(2*beta.hat$objective + 2*length(beta.hat$par),3),"]"),
      paste0("Normal [AIC: ",
             round(2*norm.hat$objective + 2*length(norm.hat$par),3),"]"),
      paste0("Gamma [AIC: ",
             round(2*gamma.hat$objective + 2*length(gamma.hat$par),3),"]"),
      paste0("Inv Gauss [AIC: ",
             round(2*invgaus.hat$objective + 2*length(invgaus.hat$par),3),"]"),
      paste0("Log-normal [AIC: ",
             round(2*lnorm.hat$objective + 2*length(lnorm.hat$par),3),"]")))+
  ggtitle("Clothing insulation level")
ggsave(paste0(figpath,"distribution_choice_1.png"),
       width = 20,
       height = 10,
       units = "cm")
#gamma is best

fit.gamma <- glm(clo ~ tOut + tInOp + sex,
                 data = c.data,
                 family = Gamma(link = "cloglog"))
add1(object = fit.gamma,
     scope = ~.+tOut:sex+tInOp:sex+tOut:tInOp,
     test = "Chisq")
fit.gamma <- update(fit.gamma, .~.+tOut:sex)
Anova(fit.gamma, type = "III", test = "LR")
add1(object = fit.gamma, scope = ~.+tInOp:sex+tOut:tInOp, test = "Chisq")
```

```
drop1(object = fit.gamma, test = "Chisq")
fit.gamma <- update(fit.gamma, .~.-tInOp)
anova(fit.gamma, test = "Chisq")
Anova(fit.gamma, type = "III", test = "LR")

#### goodness of fit:
pchisq(deviance(fit.gamma),
       df = dim(c.data)[1] - length(coefficients(fit.gamma)),
       lower.tail = F) #=> response distribution, link and eta is appropriate

summary(fit.gamma)
1 - pchisq(48.344,df=799) ## pass the goodness of fit.

confint(fit.gamma)

#Manual fit
glm.gamma.w <- function(theta){
  y <- c.data$clo

  #define the shape, k, based on the dispersion parameter theta[5]
  k <- 1/theta[5]

  #estimate eta
  eta <- theta[1] +
         theta[2] * c.data$tOut +
         theta[3] * as.numeric(c.data$sex == "male") +
         theta[4] * as.numeric(c.data$sex == "male") * c.data$tOut
  #calculate mu from eta
  mu <- 1-exp(-exp(eta))

  #calculate the negative log-likelihood
  nll <- -sum(dgamma(y, shape = k, scale = mu/k, log = T))
  return(nll)
}

#optimize parameters
manual.fit <- nlminb(start = c(0,0,0,0,1), objective = glm.gamma.w)
#calculate standard deviations through fischers information matrix
manual.sd <- sqrt(diag(solve(hessian(func = glm.gamma.w, x = manual.fit$par))))
#Get the four model parameters. The last value is the dispersion parameter
manual.fit$par
manual.sd

summary(fit.gamma)
(AIC_manuel <- manual.fit$objective*2 + 5*2)
#manual AIC is 0.03 better than glm

############################ 2) residual analysis ################################
par(mfrow=c(2,2))
postscript(file.path(getwd(), "fit.gamma.eps"),
           horizontal = FALSE,
           onefile = FALSE,
           paper = "special",
           height = 10,
           width = 10)
#png(filename = "/Users/mortenjohnsen/OneDrive - Danmarks Tekniske Universitet/DTU/10.
Semester/02424 - Advanced Dataanalysis and Statistical Modellling/02424---
Assignments/Assignment 2/residual_analysis_1.png", width = 20, height = 10, units =
"cm", res = 1000)
plot(fit.gamma, pch = 16)
dev.off()
c.data$residuals <- fit.gamma$residuals
c.data$leverage <- hatvalues(fit.gamma)


#gender-specific residual analysis
c.data$pred <- predict(fit.gamma)
```

```
c.data$pearson <- residuals(fit.gamma, type = "pearson")

sigma_sq<-fit.gamma$deviance/(dim(c.data)[1] - length(coefficients(fit.gamma)))
c.data$stdpearson <- c.data$pearson/sqrt(sigma_sq*(1-c.data$leverage))

first <- ggplot(c.data)+
  geom_point(aes(x = pred, y = residuals))+
  geom_hline(aes(yintercept = 0), colour = "blue", linetype = "dashed")+
  geom_smooth(aes(x = pred, y = residuals), colour = "blue", se = F)+
  theme_bw()+
  labs(x = "Predicted", y = "Residuals")+
  ggtitle("Residuals vs Fitted")

second <- ggplot(c.data)+
  geom_point(aes(x = pred, y = sqrt(stdpearson)))+
  geom_smooth(aes(x=pred,y = sqrt(stdpearson)), colour = "blue", se = F)+
  theme_bw()+
  labs(x = "Predicted", y = TeX("$\\sqrt{Std. Pearson Residuals}$"))+
  ggtitle("Scale-Location")

third <- ggplot(c.data, aes(sample = stdpearson))+
  stat_qq_band(fill = "blue", alpha = 0.2)+
  stat_qq_line(colour = "blue")+
  stat_qq_point()+
  theme_bw()+
  labs(x = "Theoretical quantiles", y = "Std. Pearson Residuals")+
  ggtitle("Normal QQ")

fourth <- ggplot(c.data, aes(x = leverage, y = stdpearson))+
  geom_point()+
  geom_hline(aes(yintercept = 0), colour = "blue", linetype = "dashed")+
  geom_smooth(colour = "blue", se = F)+
  theme_bw()+
  labs(x = "Leverage", y = "Std. Pearson Residuals")+
  ggtitle("Residuals vs Leverage")

grid.arrange(first, third, second, fourth, nrow = 2)

#Check whether residuals are i.i.d.
ggplot(c.data)+
  geom_boxplot(aes(x = sex, y = pearson, fill = sex))+
  theme_bw()+
  ggtitle("Residual variation difference between genders")
ggsave(filename = paste0(figpath,"gender_variance.png"),
       width = 20,
       height = 10,
       units = "cm")
#=> Residuals are not identically distributed. This will be addressed later.

par(mfrow=c(1,2))
p1 <- acf(c.data$pearson, main = "ACF pearson residuals")
p2 <- pacf(c.data$pearson, main = "PACF pearson residuals")
#=> residuals are not independent.

############################## 3) Model interpretation ##############################
plot_summs(fit.gamma)+
  ggtitle("Parameter estimates [95% CI]")
ggsave(filename = paste0(figpath,"forest1.png"),
       width = 20,
       height = 10,
       units = "cm")

summary(fit.gamma)

############### 4) Fitting the model using subjId instead of sex #################
c.data2 <- dplyr::select(clothing, -X)
c.data2 %<>%
```

```
  mutate(subjId = factor(subjId))
fit.gamma2 <- glm(clo ~ tOut + tInOp + subjId,
                  data = c.data2,
                  family = Gamma(link = "cloglog"))
anova(fit.gamma2, test = "Chisq")
Anova(fit.gamma2, type = "III")
add1(object = fit.gamma2,
     scope = ~.+I(tOut^2)+I(tInOp^2)+tOut:tInOp+tOut:subjId+tInOp:subjId,
     test = "Chisq")
fit.gamma2 <- update(fit.gamma2, .~.+tOut:subjId)
drop1(fit.gamma2, test = "Chisq")
anova(fit.gamma2, test = "Chisq")
# now all terms are significant and we see if additional terms should be added
add1(object = fit.gamma2,
     scope = ~.+I(tOut^2)+I(tInOp^2)+tOut:tInOp+tInOp:subjId,
     test = "Chisq")
fit.gamma2 <- update(fit.gamma2, .~.+tInOp:subjId)
Anova(fit.gamma2, type = "III")
drop1(fit.gamma2, test = "Chisq")
#type III anova show that tInOp is now no longer significant.
#We keep it, as it becomes significant again later
#all terms are now significant, see if we can add more information
add1(object = fit.gamma2,
     scope = 
~.+I(tOut^2)+I(tOut^2):subjId+I(tInOp^2)+I(tInOp^2):subjId+tOut:tInOp+tOut:tInOp:subjId,

     test = "Chisq")
fit.gamma2 <- update(fit.gamma2, .~.+I(tInOp^2))
Anova(fit.gamma2, type = "III")
add1(object = fit.gamma2, scope = ~.+I(tOut^2)+I(tInOp^2)+tOut:tInOp+tOut:tInOp:subjId,
test = "Chisq")
drop1(object = fit.gamma2, test = "Chisq")
#all terms are significant, no further terms can be added or removed.
Anova(fit.gamma2, type = "III")
fit.gamma2 <- update(fit.gamma2, .~.-tOut)
add1(object = fit.gamma2, scope = ~.+I(tOut^2)+I(tInOp^2)+tOut:tInOp+tOut:tInOp:subjId,
test = "Chisq")
drop1(object = fit.gamma2, test = "Chisq")
summary(fit.gamma2)

#subject id is highly significant -> normally this would be an indicator to use a mixed
model instead.
#the residual deviance for this model is a lot lower than for the sex-based model
above.

############  5) Residual analysis including within day autocorrelation ############
c.data2$pearson <- residuals(fit.gamma2, type = "pearson")
ggplot(c.data2)+
  geom_boxplot(aes(x = subjId, y = pearson, fill = sex))+
  theme_bw()+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))+
  labs(y = "Pearson residual", x = "Subject ID")
ggsave(paste0(figpath,"residual_subjid2.png"), width = 20, height = 10, units = "cm")

#still not constant residual variance as variations can be seen based on the subjId
c.data2$residuals <- fit.gamma2$residuals
c.data2$leverage <- hatvalues(fit.gamma2)
#gender-specific residual analysis
c.data2$pred <- predict(fit.gamma2)

sigma_sq <- fit.gamma2$deviance / (dim(c.data2)[1] - length(coefficients(fit.gamma2)))
c.data2$stdpearson <- c.data2$pearson/sqrt(sigma_sq*(1-c.data2$leverage))

first <- ggplot(c.data2)+
  geom_point(aes(x = pred, y = residuals))+
  geom_hline(aes(yintercept = 0), colour = "blue", linetype = "dashed")+
  geom_smooth(aes(x = pred, y = residuals), colour = "blue", se = F)+
  theme_bw()+
  labs(x = "Predicted", y = "Residuals")+
  ggtitle("Residuals vs Fitted")

second <- ggplot(c.data2)+
  geom_point(aes(x = pred, y = sqrt(stdpearson)))+
  geom_smooth(aes(x=pred,y = sqrt(stdpearson)), colour = "blue", se = F)+
  theme_bw()+
  labs(x = "Predicted", y = TeX("$\\sqrt{Std. Pearson Residuals}$"))+
  ggtitle("Scale-Location")

third <- ggplot(c.data2, aes(sample = stdpearson))+
  stat_qq_band(fill = "blue", alpha = 0.2)+
  stat_qq_line(colour = "blue")+
  stat_qq_point()+
  theme_bw()+
  labs(x = "Theoretical quantiles", y = "Std. Pearson Residuals")+
  ggtitle("Normal QQ")

fourth <- ggplot(c.data2, aes(x = leverage, y = stdpearson))+
  geom_point()+
  geom_hline(aes(yintercept = 0), colour = "blue", linetype = "dashed")+
  geom_smooth(colour = "blue", se = F)+
  theme_bw()+
  labs(x = "Leverage", y = "Std. Pearson Residuals")+
  ggtitle("Residuals vs Leverage")

png(filename = paste0(figpath,"residual_analysis_2.png"), width = 20, height = 10,
units = "cm", res = 1000)
grid.arrange(first, third, second, fourth, nrow = 2)
dev.off()

#again the residuals are fairly well-behaved as compared to the predicted values
#showing a few outliers, but all residuals are within cooks distance.
#the qqplot show poor normality. Even when accounting for 95% CIs.
qqPlot(residuals(fit.gamma2))

#looking at the within day autocorrelation to examine independency.
par(mfrow=c(1,2))
acf(residuals(fit.gamma2, type = "pearson"), main = "ACF")
pacf(residuals(fit.gamma2, type = "pearson"), main = "PACF")
#still seeing overall autocorrelation
acf <- c()
lag <- c()
subject <- c()

for (i in unique(c.data2$subjId)){ # Loop over subjects
  # Take out data for that subject
  c.data2 %>%
    filter(subjId == i) -> tmp
  for (j in unique(tmp$day)){ # Loop over different days that the subject has been at
work
    # Look at that specific day, calculate acf
    tmp %>%
      filter(day == j) %>%
      arrange(time) %>%
      dplyr::select(pearson) %>%
      acf(plot = F) -> acf.tmp
    # Save acf value for that subject on that day
    acf <- c(acf, acf.tmp$acf)
    lag <- c(lag, 0:(length(acf.tmp$acf)-1))
    subject <- c(subject, rep(i, length(acf.tmp$acf)))
  }
}

# Calculate the average per lag (accross subjects)
acf.data <- data.frame("acf" = acf, "lag" = lag, "subject" = subject)
```

```r
mean.acf.data <- acf.data %>%
  group_by(lag) %>%
  summarise(lag_avg = mean(acf))
mean.acf.data

ggplot(acf.data)+
  geom_col(aes(x = lag, y = acf, fill = subject), position = "dodge", show.legend = F)+
  theme_bw()+
  scale_fill_manual(values = rep("grey",length(unique(subject))))+
  geom_hline(aes(yintercept = qnorm(1-0.05/2)/sqrt(dim(c.data2)[1]), colour = "95%
significance level"), linetype = "dashed", linewidth = 0.8)+
  geom_hline(aes(yintercept = -qnorm(1-0.05/2)/sqrt(dim(c.data2)[1])), linetype =
"dashed", colour = "royalblue1", linewidth = 0.8)+
  geom_segment(data = mean.acf.data, aes(x = lag-0.5, xend = lag+0.5, y = lag_avg, yend
= lag_avg, colour = "ACF average pr lag"))+
  scale_x_continuous(n.breaks = 6)+
  labs(colour = "")+
  scale_colour_manual(values = c("royalblue1", "black"))+
  theme(legend.position = "top")+
  guides(colour = guide_legend(override.aes=list(linetype = c("dashed", "solid"),
linewidth = c(0.5, 0.5))))+
  ggtitle("Within day autocorrelation for each subject")
ggsave(paste0(figpath,"within_day_autocor.png"), width = 20, height = 10, units = "cm")

###################### 6) Optimal weight/dispersion parameter #####################
dummy <- dummyVars(" ~ .", data = c.data2)
new <- data.frame(predict(dummy, newdata = c.data2))

subjects <- length(names(new)[str_detect(names(new), pattern = "subjId")])-1

#Define the design matrix
X <- as.matrix(cbind(1, new$tOut, new$sexmale, new$tOut*new$sexmale))

glm.gamma.w2 <- function(theta){
  y <- new$clo
  #k <- rep(1/theta[1], length(y))
  beta <- theta[-c(1,2)]
  k <- numeric(length(y))

  k.male <- 1/theta[1]
  k.female <- 1/theta[2]
  k[as.logical(new$sexmale)] <- k.male
  k[as.logical(new$sexfemale)] <- k.female #shape

  eta <- as.numeric(X %*% matrix(beta, ncol = 1))

  mu <- 1-exp(-exp(eta))

  return(-sum(dgamma(y, shape = k, scale = mu/k, log = T)))
}

manual.fit2 <- nlminb(start = c(1,1,as.numeric(coefficients(fit.gamma))), objective =
glm.gamma.w2)

#dispersion parameter males = 1/k_male, dispersion parameter females = 1/k_female,
model parameters....
manual.fit2$par
1/manual.fit2$par[1:2]
manual.sd2 <- sqrt(diag(solve(hessian(func = glm.gamma.w2, x = manual.fit2$par))))
manual.fit2$objective
manual.sd2

#looking at the variance associated with each gender based on the weight:
theta.hat <- c(manual.fit2$par[3:6], 1/manual.fit2$par[1:2])
#for the gamma distribution we get: var = shape*scale^2 = k*theta^2
#with our parametrization: scale = mu/shape = mu/k
#Thus: var = shape * mu^2/shape^2 = mu^2/shape = mu^2/k
```

```
#(our weight w is actually an estimate of the shape/dispersion parameter k) and thus:
##male
var.male <- (theta.hat[1] + theta.hat[2] * c.data$tOut[c.data$sex == "male"] +
theta.hat[3] * as.numeric(c.data$sex == "male")[c.data$sex == "male"] + theta.hat[4] *
as.numeric(c.data$sex == "male")[c.data$sex == "male"] * c.data$tOut[c.data$sex ==
"male"])^2 / theta.hat[5]
mean(var.male)
##female
var.female <- (theta.hat[1] + theta.hat[2] * c.data$tOut[c.data$sex == "female"] +
theta.hat[3] * as.numeric(c.data$sex == "female")[c.data$sex == "female"] +
theta.hat[4] * as.numeric(c.data$sex == "female")[c.data$sex == "female"] *
c.data$tOut[c.data$sex == "female"])^2 / theta.hat[6]
mean(var.female)
par(mfrow=c(1,1))
plot(var.female, type = "l")
lines(var.male, col = 3)
#Higher variance for the women as expected

#Notes on the variance function V(.) as compareed to the variance Var(.) on page 93.

######################### 7) Profile likelihood ############################
glm.gamma.w.pf <- function(d_male,d_female){
  y <- new$clo
  k <- numeric(length(y))
  k.male <- 1/d_male
  k.female <- 1/d_female
  k[as.logical(new$sexmale)] <- k.male
  k[as.logical(new$sexfemale)] <- k.female #shape

  tmp.func <- function(beta, shape = k){
    eta <- as.numeric(X %*% matrix(beta, ncol = 1))

    mu <- 1-exp(-exp(eta))
    return(-sum(dgamma(y, shape = shape, scale = mu/shape, log = T)))
  }
  fit.tmp <- nlminb(start = c(0,0,0,0), objective = tmp.func)
  return(fit.tmp$objective)
}


w1 <- seq(0.020, 0.050, length.out = 60)
w2 <- seq(0.045, 0.12, length.out = 60)
z <- outer(w1, w2, FUN = Vectorize(function(w1,w2) glm.gamma.w.pf(w1,w2)))

png(filename = "countour_pf.png", width = 20, height = 10, units = "cm", res = 500)
contour(w1, w2, z, xlab = "male", ylab = "female", nlevels = 50)
dev.off()

glm.gamma.w.pf2 <- function(d){
  d_male <- d[1]
  d_female <- d[2]
  y <- new$clo
  k <- numeric(length(y))
  k.male <- 1/d_male
  k.female <- 1/d_female
  k[as.logical(new$sexmale)] <- k.male
  k[as.logical(new$sexfemale)] <- k.female #shape

  tmp.func <- function(beta, shape = k){
    eta <- as.numeric(X %*% matrix(beta, ncol = 1))

    mu <- 1-exp(-exp(eta))
    return(-sum(dgamma(y, shape = shape, scale = mu/shape, log = T)))
  }
  fit.tmp <- nlminb(start = c(0,0,0,0), objective = tmp.func)
  return(fit.tmp$objective)
}
```

```r
gg_plot_data <- tidyr::expand(data.frame(w1,w2), w1, w2)

gg_plot_data$z <- apply(gg_plot_data, MARGIN = 1, glm.gamma.w.pf2)

ggplot(gg_plot_data, aes(x = w1, y = w2, z = z))+
  #geom_raster(aes(fill = -z), interpolate = T, alpha = 0.5)+
  geom_contour(bins = 30, colour = "black")+
  geom_point(aes(x = manual.fit2$par[1], y = manual.fit2$par[2], colour = "Optimal"))+
  scale_colour_manual(values = c("black"))+
  #scale_fill_gradient(low = "white", high = "black")+
  theme_minimal()+
  labs(colour = "",
       x = "Male dispersion parameter",
       y = "Female dispersion parameter",
       fill = "log-likelihood")+
  theme(legend.position = "right")+
  ggtitle("Profile likelihood contours for gender-specific dispersion parameters")
ggsave("contour.png", width = 20, height = 14, unit = "cm")
```