

```

##Island Colonization Analysis
##Ethan Cochrane, 7 December, 2020

## clear memory and set proper directory

##reading and checking data
All_dat<- read.csv("JWP_master_Nov2020.csv")
island_dat<- na.omit(All_dat) #omitting NAs as can't use in glm, but
there are methods to deal with this.
str(island_dat)
summary(island_dat)

#graphically explore data
attach(island_dat)
par(mfrow=c(2,3))
logsize <- log(island_dat$Size)
plot(logsize, island_dat$BP, ylab = "BP", xlab = "Log(Size)", pch=16)
plot(island_dat$Latitude, island_dat$BP, ylab = "BP", xlab =
"Latitude",pch=16)
plot(island_dat$Elevation, island_dat$BP, ylab = "BP", xlab =
"Elevation",pch=16)
plot(island_dat$Coastal_Index, island_dat$BP, ylab = "BP", xlab =
"Coastal_Index",pch=16)
plot(island_dat$Isolation_Index, island_dat$BP, ylab = "BP", xlab =
"Isolation_Index",pch=16)
boxplot(island_dat$BP ~ island_dat$Sail, data = island_dat, ylab = "BP",
xlab = "Sail")

#examine bi-variable correlations
cor.test(logsize, BP, method="kendall")
cor.test(Latitude, BP, method="kendall")
cor.test(Elevation, BP, method="kendall")
cor.test(Coastal_Index, BP, method="kendall")
cor.test(Isolation_Index, BP, method = "kendall")

#Mann-Whitney U test if BP linked to sail
sail_y <- subset(island_dat, island_dat$Sail == "y")
hist(sail_y$BP)
sail_n <- subset(island_dat, island_dat$Sail == "n")
hist(sail_n$BP)
wilcox.test(BP~Sail, data = island_dat)

#estimate probability density function of response variable BP to choose
appropriate model

hist(BP)
plot(density(BP))

#choose a gamma distribution

#setting up models
Gmodel1 <- glm(BP ~ Size, family = Gamma())
Gmodel2 <- glm(BP ~ Size+Latitude, family = Gamma())
Gmodel3 <- glm(BP ~ Size+Latitude+Elevation,family = Gamma())
Gmodel4 <- glm(BP ~ Size+Latitude+Elevation+Coastal_Index, family =
Gamma())

```

```

Gmodel5 <- glm(BP ~
Size+Latitude+Elevation+Coastal_Index+Isolation_Index, family = Gamma())
Gmodel6 <- glm(BP ~ Latitude, family = Gamma())
Gmodel7 <- glm(BP ~ Latitude+Elevation, family = Gamma())
Gmodel8 <- glm(BP ~ Latitude+Elevation+Coastal_Index, family = Gamma())
Gmodel9 <- glm(BP ~ Latitude+Elevation+Coastal_Index+Isolation_Index,
family = Gamma())
Gmodel10 <- glm(BP ~ Elevation, family = Gamma())
Gmodel11 <- glm(BP ~ Elevation+Coastal_Index, family = Gamma())
Gmodel12 <- glm(BP ~ Elevation+Coastal_Index+Isolation_Index, family =
Gamma())
Gmodel13 <- glm(BP ~ Coastal_Index, family = Gamma())
Gmodel14 <- glm(BP ~ Coastal_Index+Isolation_Index, family = Gamma())
Gmodel15 <- glm(BP ~ Isolation_Index, family = Gamma())
Gmodel16 <- glm(BP ~ Size+Elevation, family = Gamma())
Gmodel17 <- glm(BP ~ Size+Elevation+Coastal_Index, family = Gamma())
Gmodel18 <- glm(BP ~ Size+Elevation+Coastal_Index+Isolation_Index, family
= Gamma())
Gmodel19 <- glm(BP ~ Size+Coastal_Index, family = Gamma())
Gmodel20 <- glm(BP ~ Size+Coastal_Index+Isolation_Index, family =
Gamma())
Gmodel21 <- glm(BP ~ Size+Isolation_Index, family = Gamma())
Gmodel22 <- glm(BP ~ Size+Latitude+Coastal_Index, family = Gamma())
Gmodel23 <- glm(BP ~ Size+Latitude+Coastal_Index+Isolation_Index, family
= Gamma())
Gmodel24 <- glm(BP ~ Size+Latitude+Isolation_Index, family = Gamma())
Gmodel25 <- glm(BP ~ Latitude+Coastal_Index+Isolation_Index, family =
Gamma())
Gmodel26 <- glm(BP ~ Elevation+Isolation_Index, family = Gamma())

#evaluating models
library (MuMIn)

MS_AIC_G <- model.sel(Gmodel1, Gmodel2, Gmodel3, Gmodel4, Gmodel5,
Gmodel6, Gmodel7, Gmodel8, Gmodel9, Gmodel10,
Gmodel11, Gmodel12, Gmodel13, Gmodel14, Gmodel15,
Gmodel16, Gmodel17, Gmodel18, Gmodel19,
Gmodel20, Gmodel21, Gmodel22, Gmodel23, Gmodel24,
Gmodel25, Gmodel26, rank = AIC)

MS_BIC_G <- model.sel(Gmodel1, Gmodel2, Gmodel3, Gmodel4, Gmodel5,
Gmodel6, Gmodel7, Gmodel8, Gmodel9, Gmodel10,
Gmodel11, Gmodel12, Gmodel13, Gmodel14, Gmodel15,
Gmodel16, Gmodel17, Gmodel18, Gmodel19,
Gmodel20, Gmodel21, Gmodel22, Gmodel23, Gmodel24,
Gmodel25, Gmodel26, rank = BIC)

MS_AIC_G
MS_BIC_G

#use step-wise model selection
library(MASS)
stepAIC(Gmodel15)

#inspect best fitting model
summary(Gmodel15)
plot(Gmodel15)

```

```
#plot fitted model
range(Isolation_Index)
xII <- seq(0,150, 5)
yII <- predict(Gmodell15, list(Isolation_Index=xII), type="response",
se.fit = F)
plot(Isolation_Index, BP, pch=16, xlab="Isolation Index",
ylab="Colonization BP")
lines(xII, yII, col="red", lwd=2)

library(jtools)
effect_plot(Gmodell15, pred = Isolation_Index, interval=T, plot.points=T)
```