

IS figures

Maya Lapp

1/8/2020

```
setwd("~/Documents/WoosterStuff/fall2019/IS_organized/analysis")  
source(file="setupFile2.R")  
setwd("~/Documents/WoosterStuff/fall2019/IS_organized/analysis")  
source(file="setupVallino.R")
```

default graph settings

```

color="Set2"
axlab_size <- 12
axtick_size <- 10

ax_kf = expression(paste(bolditalic("final-institution"))))
axlab_bm<-"Biomass (% remaining)"
axlab_density <- "Density"
axlab_time <- "Period"
axlab_perc <- "Percent remianing"

# make graph look nice by adding this to the end
setGraphs<- theme_bw()+
  theme(axis.text=element_text(size=axtick_size, color="black"),
        axis.title=element_text(size=axlab_size,face="bold", color="black"), legend.text
= element_text(size=axtick_size))+
  theme(panel.border = element_blank(), axis.line = element_line(colour = "black"))

# scale_color_stata()
# scale_color_brewer(palette="Set1")
#colorDiscrete = scale_color_brewer(palette="Dark2")
colorDiscrete = scale_color_brewer(palette="Set2")
colorCont = scale_color_viridis_c(option = "magma")
#fillDiscrete = scale_fill_brewer(palette="Dark2")
fillDiscrete = scale_fill_brewer(palette="Set2")
fillCont = scale_fill_viridis_c(option = "magma")

# first two colors in Dark2
#1b9e77
#d95f02

color1="#66c2a5"
color2="#fc8d62"
color3="#8da0cb"

# scale_color_colorblind()
# scale_color_OkabeIto()

point_size = 3
line_size = 1.2

```

pred prey ABM example

```

setwd("~/Documents/WoosterStuff/fall2019/IS_organized/analysis/allData_IS")
sheep50wolves100<- read.csv(file="sheep50wolves100.csv", stringsAsFactors = FALSE)
sheep50wolves100<-as.data.frame(sheep50wolves100)[1:500,]
sheep50wolves100$time<- 1:500
for(i in 1:20){
  sheep50wolves100[,i]<- as.numeric(sheep50wolves100[,i])
}

```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

```
## Warning: NAs introduced by coercion
```

```

sheep50wolves100<- sheep50wolves100%>%gather(key="run", value="numberTurtles", -time )%
>%mutate(numberTurtles=as.numeric(numberTurtles))

sheep50wolves100<- sheep50wolves100%>%separate(run, into = c("Agent", "run"))

v1<-sheep50wolves100%>%filter(time<376)%>%ggplot(aes(x=time, y=numberTurtles))+
  geom_path(aes(color=Agent, alpha=run), size=line_size*0.8)+
  ylim(0, 20000)+
  labs(x="", y=element_blank(), color="Agent type", linetype="Run")+
  setGraphs+
  colorDiscrete+
  theme(legend.position = "none")

#####
sheep50wolves200<- read.csv(file="sheep50wolves200.csv", stringsAsFactors = FALSE)
sheep50wolves200<-as.data.frame(sheep50wolves200)[1:200,]
sheep50wolves200$time<- 1:200
for(i in 1:20){
  sheep50wolves200[,i]<- as.numeric(sheep50wolves200[,i])
}
sheep50wolves200<- sheep50wolves200%>%gather(key="run", value="numberTurtles", -time )%
>%mutate(numberTurtles=as.numeric(numberTurtles))

sheep50wolves200<- sheep50wolves200%>%separate(run, into = c("turtle", "run"))

v2<-sheep50wolves200%>%ggplot()+
  #geom_path(aes(x=time, y=numberTurtles, color=turtle, linetype=run), size=line_size*
0.8)+
  geom_path(aes(x=time, y=numberTurtles, color=turtle, alpha=run),size=line_size*0.8)+
  labs(x="", y="", color="Agent type", linetype="Run")+
  setGraphs+
  colorDiscrete+
  theme(legend.position = "none")

#####
sheepWolves_deq <-read.csv(file="wolvesSheep_difEQdata.csv", stringsAsFactors = FALSE)
sheepWolves_deq<- sheepWolves_deq[(1:550)*500,] #280,000
sheepWolves_deq<- sheepWolves_deq%>%gather(key="run", value="population", -time )%>%muta
te(population=as.numeric(population))
sheepWolves_deq<- sheepWolves_deq%>%separate(run, into = c("agent", "run"))

v3<-sheepWolves_deq%>%filter(run==100)%>%ggplot()+
  geom_path(aes(x=time, y=population, color=agent), size=line_size*0.8)+
  labs(x=element_blank(), y=element_blank(), color="Agent type", linetype="Run")+
  setGraphs+
  colorDiscrete+
  theme(legend.position = "none")

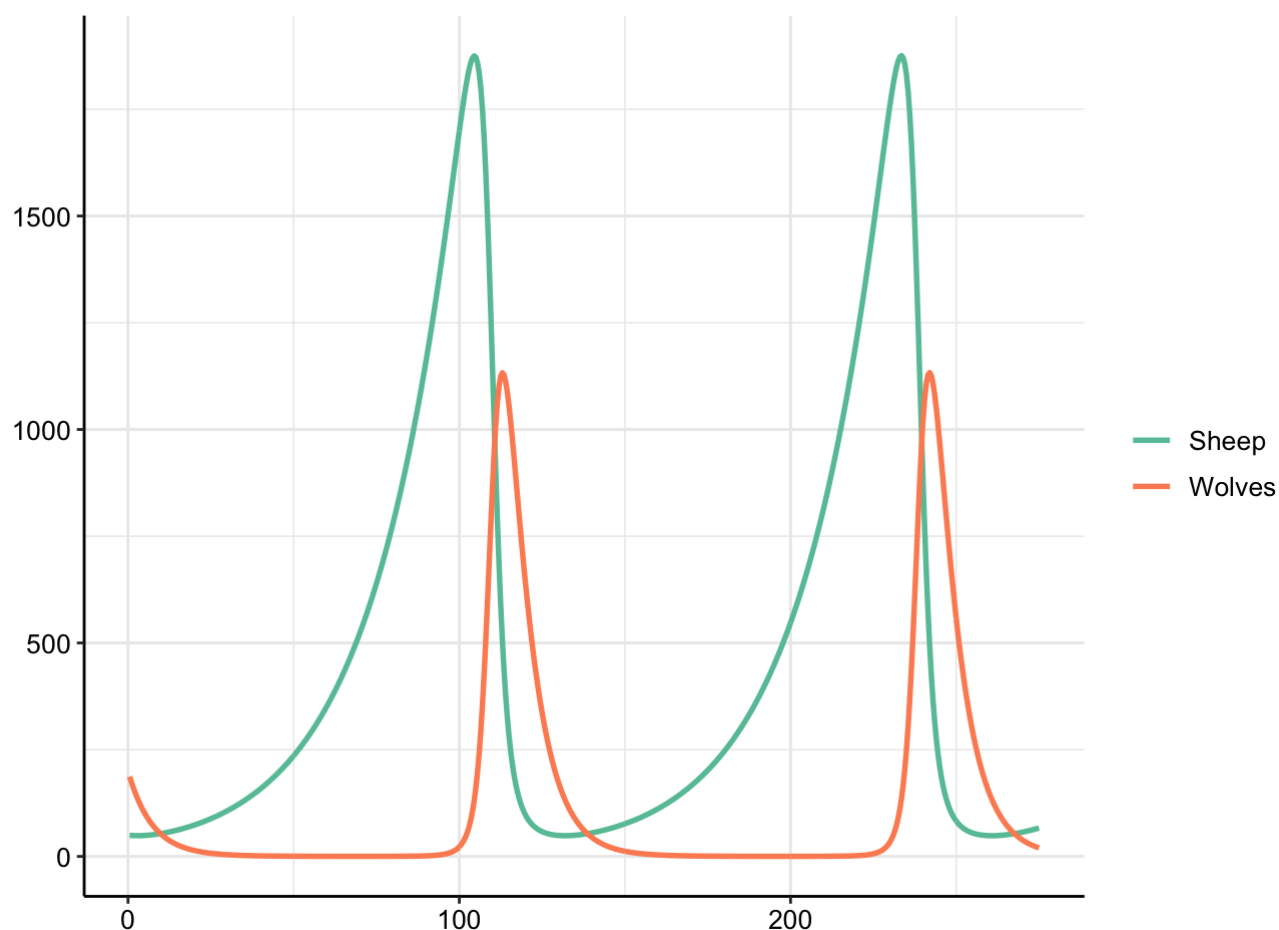
v4<-sheepWolves_deq%>%filter(run==200)%>%ggplot()+
  geom_path(aes(x=time, y=population, color=agent), size=line_size*0.8)+
  labs(x=element_blank(), y="", color=NULL, linetype="Run")+
  setGraphs+

```

```

theme(legend.text=element_text(size=axtick_size))+
colorDiscrete
v4

```



```

# arranging in grid

#extract legend
#https://github.com/hadley/ggplot2/wiki/Share-a-legend-between-two-ggplot2-graphs
g_legend<-function(a.gplot){
  tmp <- ggplot_gtable(ggplot_build(a.gplot))
  leg <- which(sapply(tmp$grobs, function(x) x$name) == "guide-box")
  legend <- tmp$grobs[[leg]]
  return(legend)}

mylegend<-g_legend(v4)

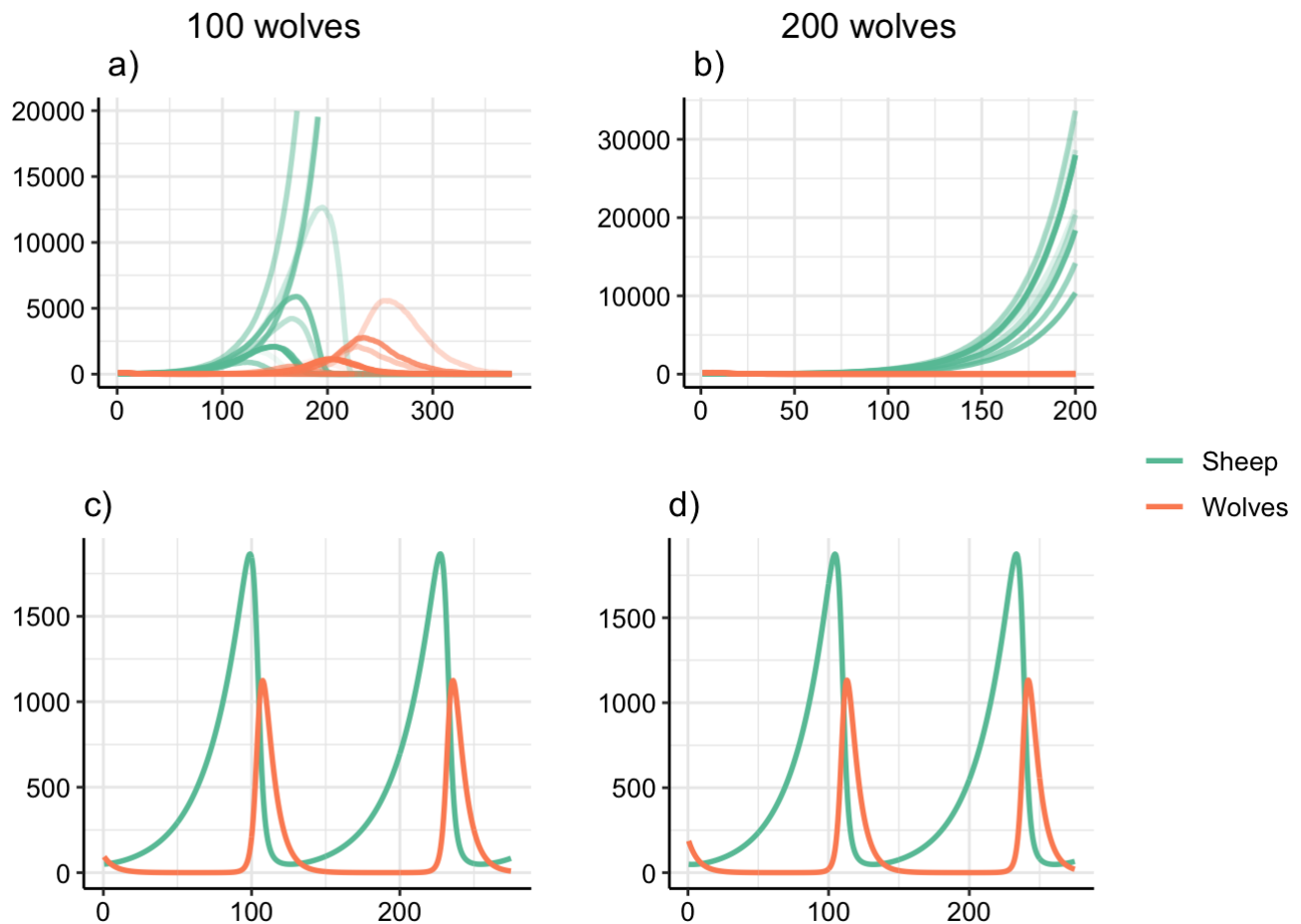
viz_deq=grid.arrange(arrangeGrob(v1+ggtitle("100 wolves \n a"),v2+ggtitle("
200 wolves \n b"),v3+ggtitle("c"),v4+ggtitle("d")) + theme(legend.position="none"),nro
w=2),mylegend, nrow=1,widths=c(9.5, 1.8))

```

```
## Warning: Using alpha for a discrete variable is not advised.
```

```
## Warning: Removed 898 rows containing missing values (geom_path).
```

```
## Warning: Using alpha for a discrete variable is not advised.
```



```
viz_deq=annotate_figure(viz_deq,  
                        bottom = text_grob(expression(bold(Time))), vjust = 0, size = axlab_size,  
                        left = text_grob(expression(bold("Population size")), size = axlab_size,  
                        rot = 90)  
                        )  
  
ggsave("viz_deq.pdf", viz_deq, width=5.75, height=4, units="in")
```

bravo model run

```

setwd("~/Documents/WoosterStuff/fall2019/IS_organized/analysis/allData_IS")
bravo_data<- read.csv("bravo_modelRun.csv", stringsAsFactors = FALSE)[1:200,]%>%mutate(T
otalBM=TotalBM/TotalBM[1])

ax_k = expression(paste(bolditalic("current-institution"))))
v1b<-bravo_data%>%ggplot(aes(x=period))+
  geom_path(aes(y=K), size=line_size)+
  labs(x=element_blank(), y=ax_k)+
  geom_vline(xintercept = 19, linetype="dashed", color="dimgrey")+
  ylim(0,20)+
  setGraphs

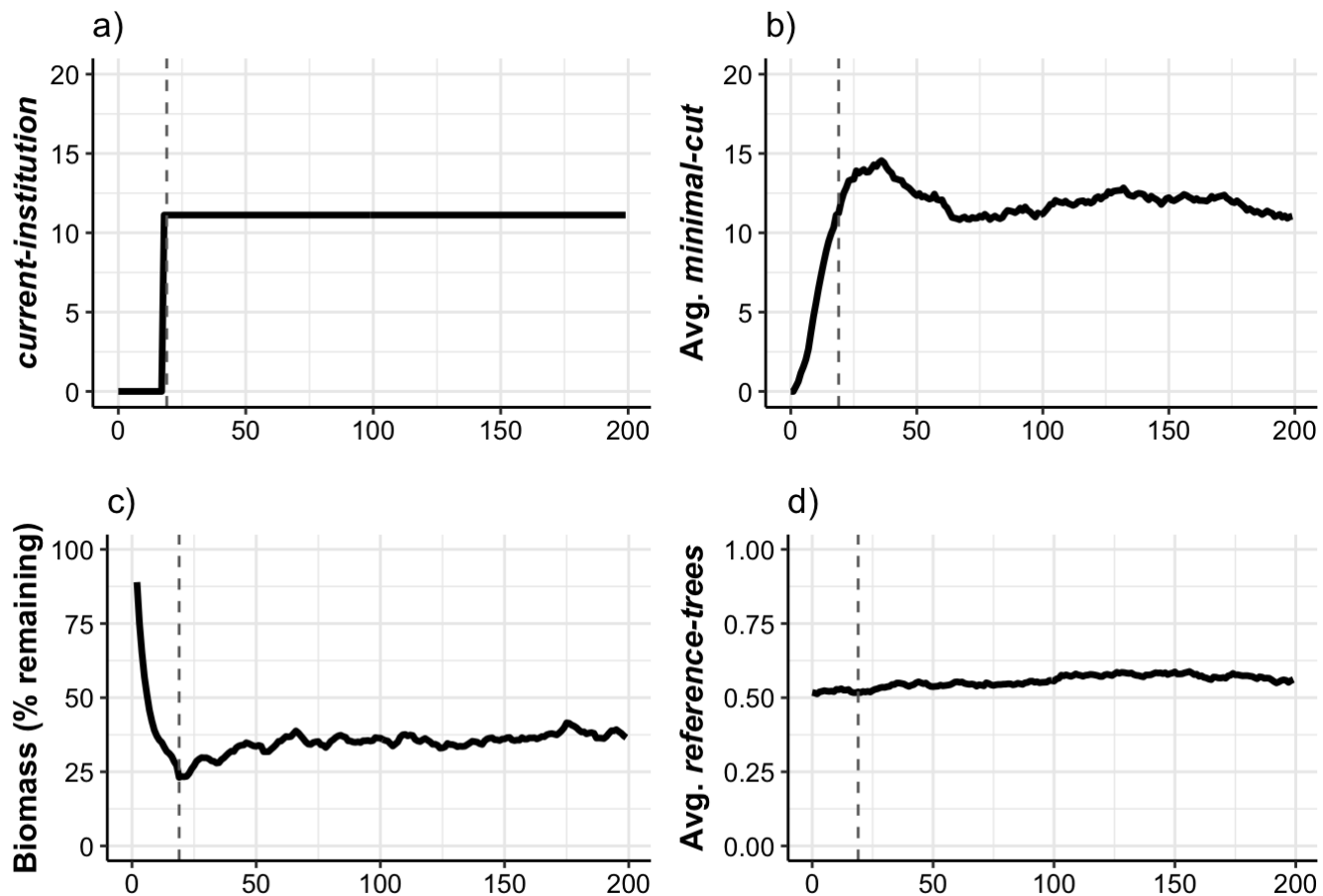
ax = expression(bold(paste("Avg. ", bolditalic("minimal-cut"))))
v2b<-bravo_data%>%ggplot(aes(x=period))+
  geom_path(aes(y=avgMinCut), size=line_size)+
  geom_vline(xintercept = 19, linetype="dashed", color="dimgrey")+
  labs(x=element_blank(), y=ax)+
  ylim(0,20)+
  setGraphs

ax = expression(bold(paste("Avg. ", bolditalic("reference-trees"))))
v3b<-bravo_data%>%ggplot(aes(x=period))+
  geom_path(aes(y=Reference.trees), size=line_size)+
  geom_vline(xintercept = 19, linetype="dashed", color="dimgrey")+
  labs(x="", y=ax)+
  ylim(0,1)+
  setGraphs

v4b<-bravo_data%>%ggplot(aes(x=period))+
  geom_path(aes(y=TotalBM*100), size=line_size)+
  geom_vline(xintercept = 19, linetype="dashed", color="dimgrey")+
  labs(x="", y=axlab_bm)+
  ylim(0,100)+
  setGraphs

viz_bravo=grid.arrange(v1b+ggtitle("a"), v2b+ggtitle("b"), v4b+ggtitle("c"), v3b+ggtitle("d"), nrow=2)

```



```
viz_bravo=annotate_figure(viz_bravo,
                           bottom = text_grob(expression(bold(Period))), vjust = 0, size = axlab_size),
)
ggsave("viz_bravo.pdf", viz_bravo, width=5.75, height=4, units="in")
```

Original Vallino model analyss

base parameters density plot

```
viz_baseVallino = baseData_vallino%>%ggplot(aes(x=TotalBM*100, fill = "A"))+
  geom_density(alpha=0.3)+
  labs(x=axlab_bm, y=axlab_density)+
  xlim(0,100)+
  setGraphs+
  theme(legend.position = "none")+
  fillDiscrete+
  geom_vline(xintercept = 26.3, linetype="dotted")

median(baseData_vallino$TotalBM)
```

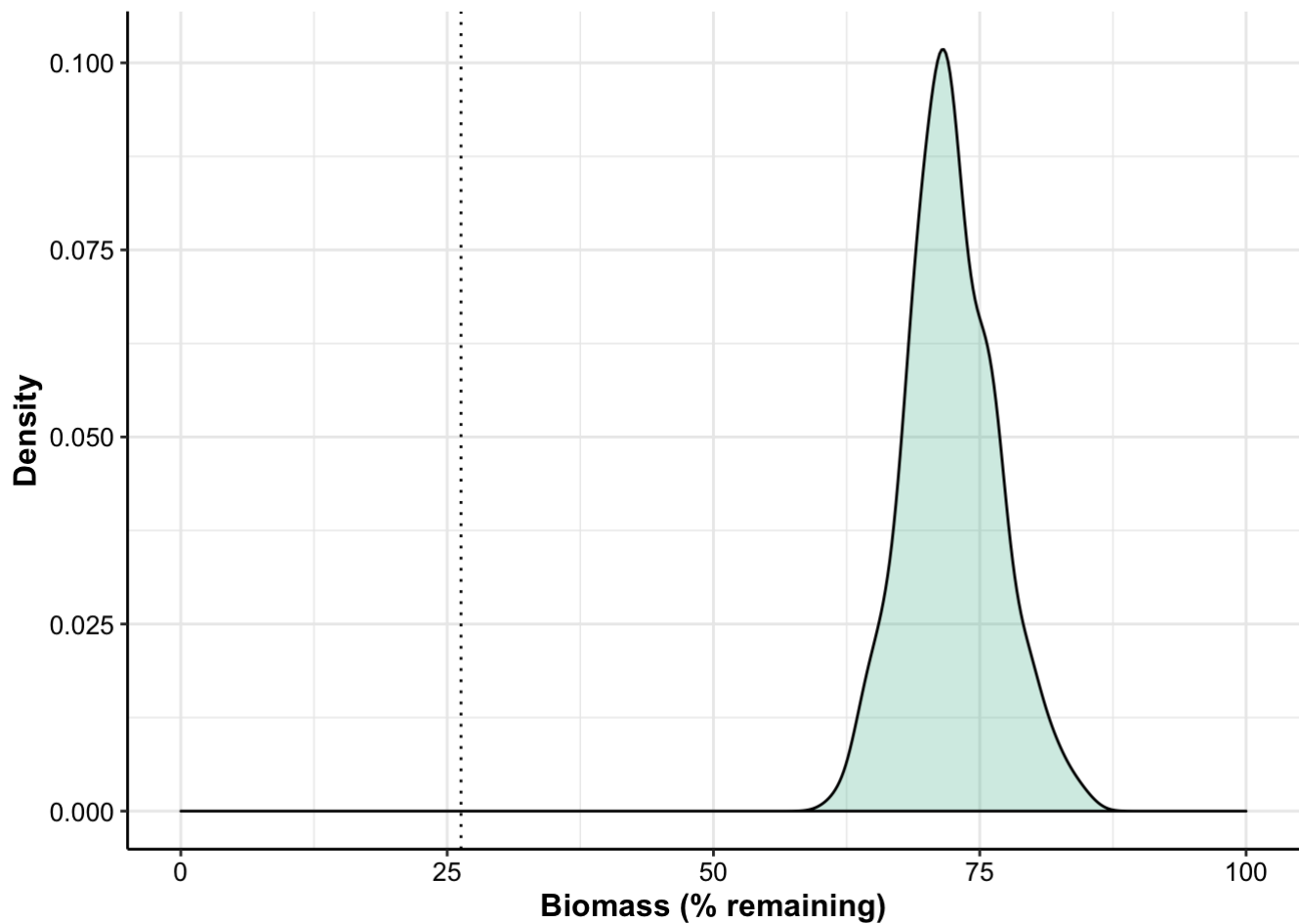
```
## [1] 0.7205842
```



```
m<-mean(baseData_vallino$TotalBM)
sdev<-sd(baseData_vallino$TotalBM)
m-sdev*1.5
```

```
## [1] 0.6607837
```

```
viz_baseVallino
```



```
viz_baseVallino=annotate_figure(viz_baseVallino,
                                top = text_grob("26.3%", vjust = 1, hjust=1.5, size = 10),
                                )

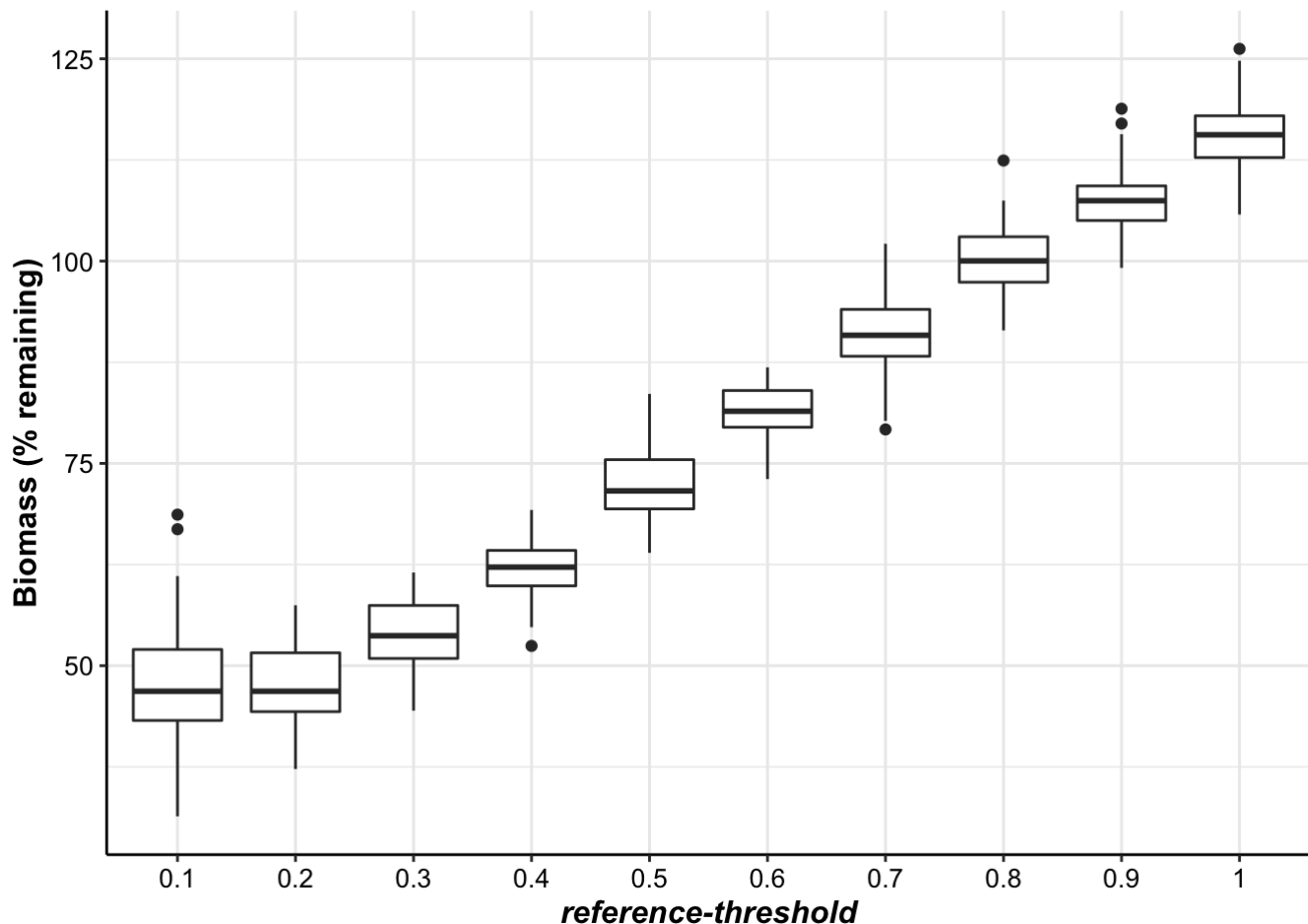
ggsave("viz_baseVallino.pdf", viz_baseVallino, width=4, height=2.5, units="in")
```

reference-threshold boxplot

```

ax_rt=expression(bolditalic("reference-threshold"))
viz_vallinoRTboxplot=rtData_vallino%>%ggplot(aes(x=as.factor(as.numeric(variableValue)/10), y=TotalBM*100))+geom_boxplot()+
  labs(x=ax_rt, y=axlab_bm)+
  setGraphs+
  theme(legend.position = "none")+
  theme(axis.title=element_text(size=axlab_size))
viz_vallinoRTboxplot

```



```

ggsave("viz_vallinoRTboxplot.pdf", viz_vallinoRTboxplot, width=4, height=2.5, units="in"
)

```

enforcement boxplot

```

ax_surveillance=expression(bold(bolditalic("surveillance-level")))
enfGraph=enfData_vallino%>%mutate(enfValue=as.factor((as.numeric(variableValue)-1)/10))
viz_vallinoEnf=enfGraph%>%ggplot(aes(x=rev(enfValue), y=TotalBM*100))+geom_boxplot()+
  labs(y=axlab_bm, x=ax_surveillance)+
  setGraphs

ggsave("viz_vallinoEnf.pdf", viz_vallinoEnf, width=4, height=2.5, units="in")

```

vallino Kf vs BM scatter plot

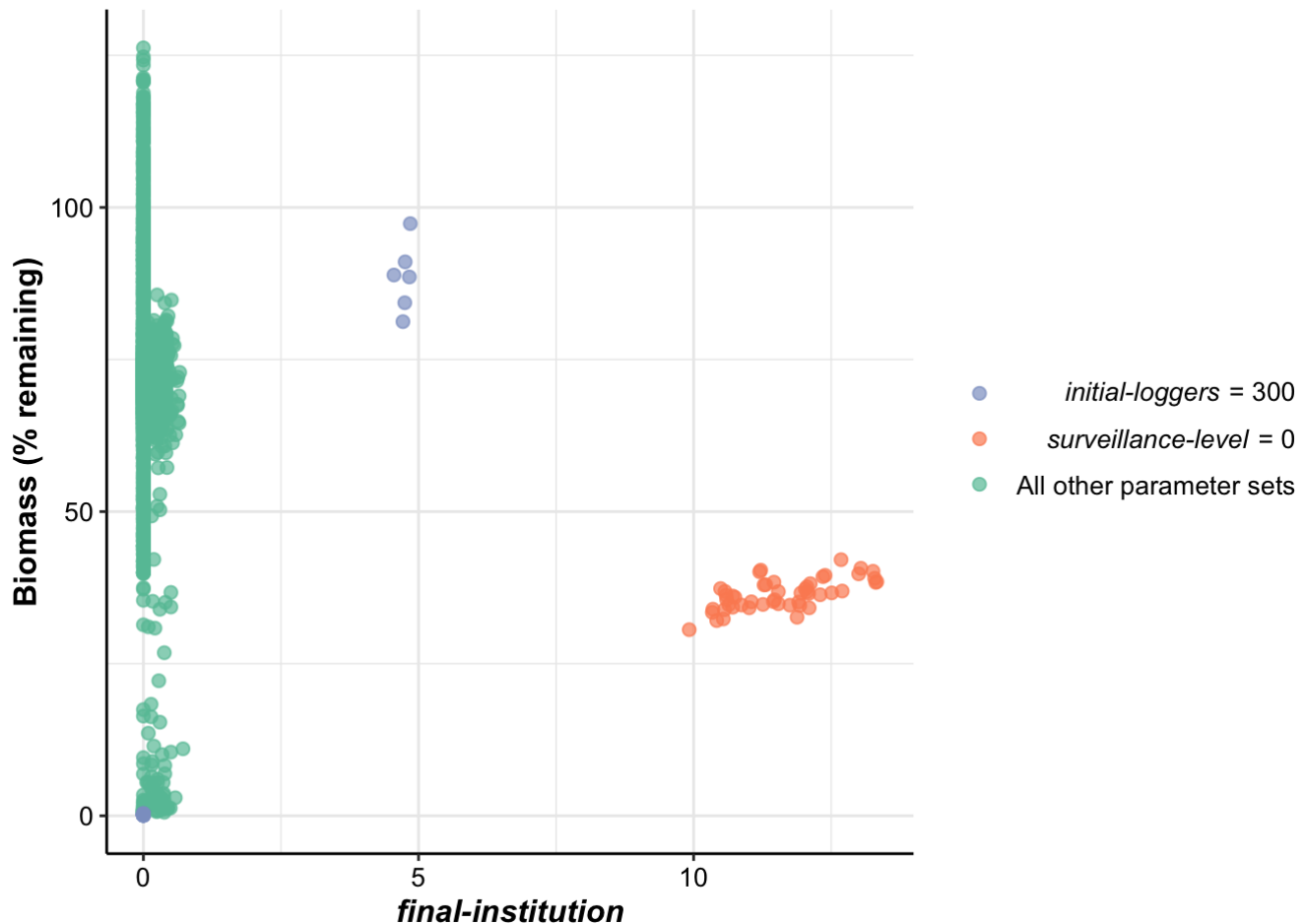
```

allData1_vallino<-allData_vallino%>%mutate(categories=ifelse(variableValue==300, "log300",
  ifelse(param=="enf" & variableValue==100, "enf100", 0)))
# with all of the data runs, situations with higher K have higher BM%
# simulations with different #loggers deviates slightly from this trend

viz_vallinoKf=allData1_vallino%>%ggplot(aes(K, TotalBM*100, color=categories))+
  geom_point(alpha=0.7, size=2)+
  setGraphs+
  labs(x=ax_kf, y=axlab_bm)+
  scale_color_brewer(palette="Set2", name="test", breaks=c("log300", "enf100", "0"), labels=c(expression(paste(italic("initial-loggers"), " = 300")), expression(paste(italic("surveillance-level"), " = 0")), "All other parameter sets"))+
  #colorDiscrete+
  theme(legend.title=element_blank())

viz_vallinoKf

```



```

ggsave("viz_vallinoKf.pdf", viz_vallinoKf, width=5.75, height=3, units="in")

```

Fit model to LV equations - not using this

```

source(file="setupLVEquations.R")

```

```
##  
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':  
##  
## smiths
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 1 rows [1].
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 1 rows [1].
```

```
## [1] 215  
## [1] 663  
## [1] 1111  
## [1] 1559  
## [1] 2007  
## [1] 2455  
## [1] 2903  
## [1] 3351
```

```

testData = read.csv("log1000.csv")
testData = testData%>%mutate(cummulativeCheaters=cumsum(numCheaters))

testData= testData%>%mutate(time=1:2001)

# ABM log=1000; data only
viz_removalLoggers=testData[1:700,]%>%ggplot(aes(x=time))+
  labs(x="Time",y= axlab_perc )+
  geom_path(aes(y=TotalBM/TotalBM[1]*100, color = "Biomass"),size=line_size)+
  geom_path(aes(y = numLoggers/numLoggers[1]*100, color = "Loggers remaining"),size=line_size)+
  geom_path(aes(y = cummulativeCheaters/numLoggers[1]*100, color = "Loggers removed"),size=line_size)+
  setGraphs+
  theme(legend.title = element_blank())+
  colorDiscrete
ggsave("viz_removalLoggers.pdf", viz_removalLoggers, width=5.5, height=2.5, units="in")

# changed ABM to make it fit LV equations
viz_abmFitLV=lvTest1%>%slice(1:200)%>%ggplot(aes(x=time))+
  geom_path(aes(x= 1:200, y=TotalBM/TotalBM[1]*100, color="Biomass"), size=line_size)+
  geom_path(aes(x=1:200, y = numLoggers/numLoggers[1]*100, color="Loggers"), size=line_size)+
  geom_path(data=fit_model_to_LV, aes(y=X/xmax*100), color="#66c2a5",size=line_size, linetype="dotted")+
  geom_path(data=fit_model_to_LV, aes(y=Y/ymax*100), color="#fc8d62", size=line_size, linetype="dotted")+
  labs(x="Time", y=axlab_perc)+
  ylim(0,150)+
  xlim(0,200)+
  setGraphs+
  theme(legend.title = element_blank())+
  colorDiscrete

lv_legend=g_legend(viz_abmFitLV)

```

```
## Warning: Removed 43 rows containing missing values (geom_path).
```

```
## Warning: Removed 3227 rows containing missing values (geom_path).
```

```
## Warning: Removed 2650 rows containing missing values (geom_path).
```

```

viz_abmFitLV=viz_abmFitLV+theme(legend.position = "none")

# compared actual ABM to LV equations
viz_abmCompareLV=compare_model_to_LV%>%ggplot(aes(x=time))+
  geom_path(aes(y=X/xmax*100), color="#66c2a5", size=line_size, linetype="dotted")+
  geom_path(aes(y=Y/ymax*100), color="#fc8d62", size=line_size, linetype="dotted")+
  geom_path(data = logEnfBM, aes(x= 1:700, y=bm.8.1/bm.8.1[1]*100, color = "Biomass"), s
size=line_size)+
  geom_path(data=logEnfLog, aes(x=1:700, y = log.8.1/log.8.1[1]*100, color = "Loggers"),
size=line_size)+
  ylim(0,150)+xlim(0,450)+
  labs(x="Time", y=axlab_perc)+
  setGraphs+
  theme(legend.position = "none")+
  colorDiscrete

viz_abm_lv=grid.arrange(arrangeGrob(viz_abmFitLV, lv_legend, ncol=2), viz_abmCompareLV)

```

```
## Warning: Removed 43 rows containing missing values (geom_path).
```

```
## Warning: Removed 3227 rows containing missing values (geom_path).
```

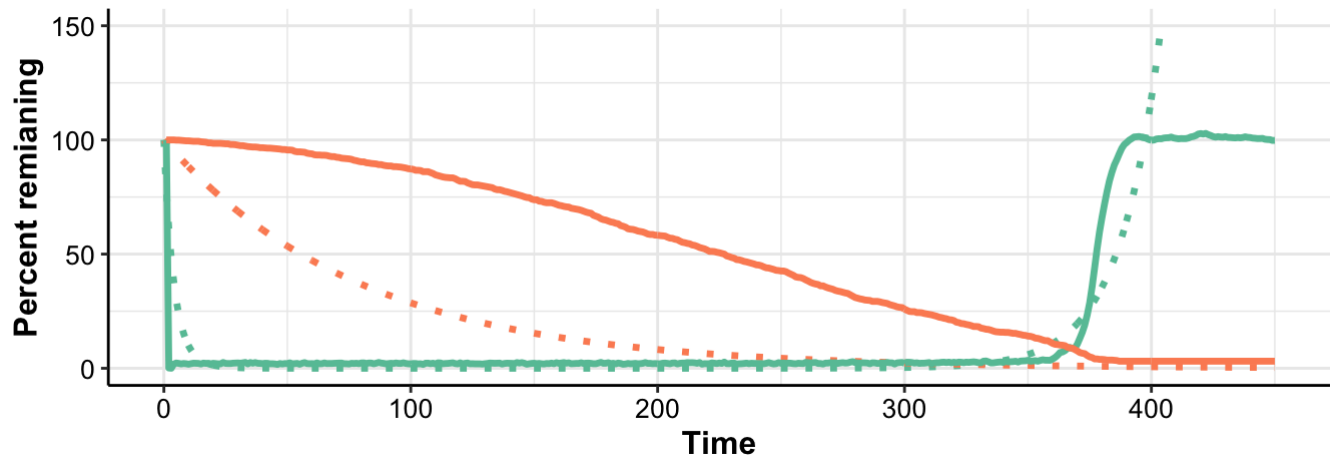
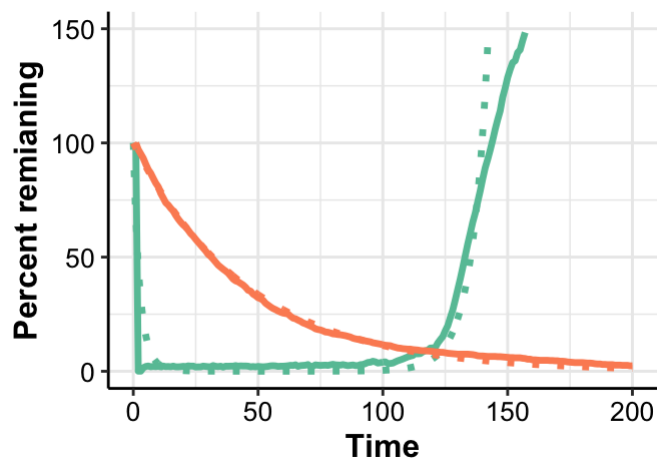
```
## Warning: Removed 2650 rows containing missing values (geom_path).
```

```
## Warning: Removed 611 rows containing missing values (geom_path).
```

```
## Warning: Removed 150 rows containing missing values (geom_path).
```

```
## Warning: Removed 250 rows containing missing values (geom_path).
```

```
## Warning: Removed 250 rows containing missing values (geom_path).
```



```
ggsave("viz_abm_lv.pdf", viz_abm_lv, width = 5.75, height = 5, units="in")
grid.arrange(viz_abmFitLV, viz_abmCompareLV)
```

```
## Warning: Removed 43 rows containing missing values (geom_path).
```

```
## Warning: Removed 3227 rows containing missing values (geom_path).
```

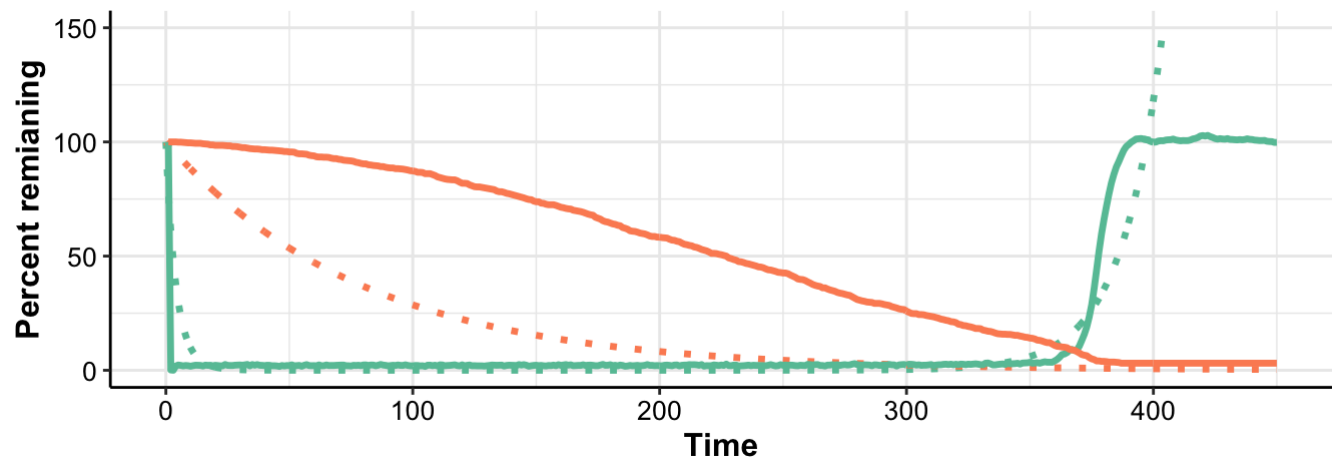
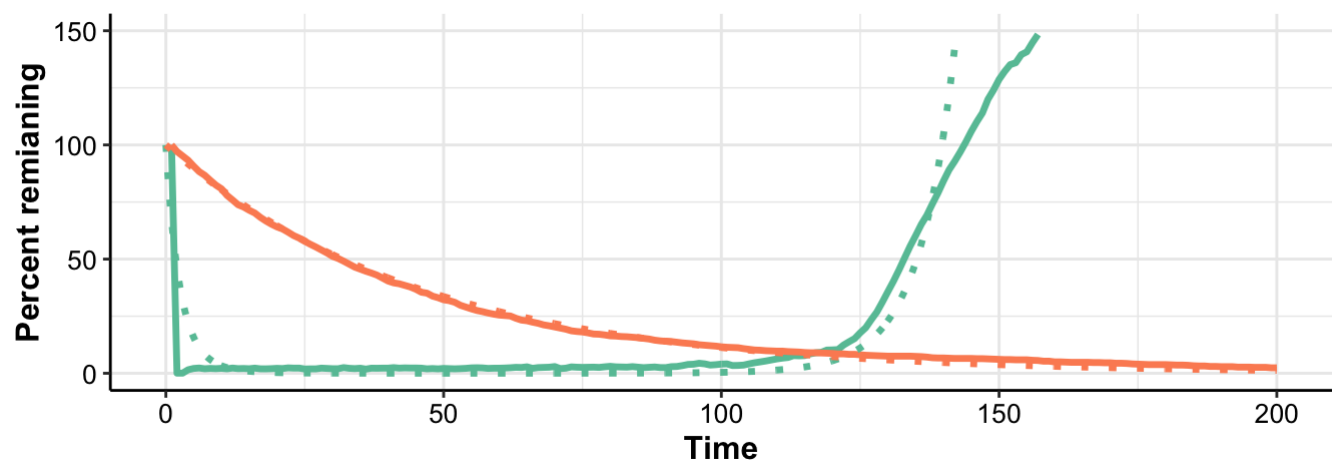
```
## Warning: Removed 2650 rows containing missing values (geom_path).
```

```
## Warning: Removed 611 rows containing missing values (geom_path).
```

```
## Warning: Removed 150 rows containing missing values (geom_path).
```

```
## Warning: Removed 250 rows containing missing values (geom_path).
```

```
## Warning: Removed 250 rows containing missing values (geom_path).
```



fit model to lv equations - using this version

```
source(file="setupLVEquations.R")
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 1 rows [1].
```

```
## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 1 rows [1].
```

```
## [1] 215
## [1] 663
## [1] 1111
## [1] 1559
## [1] 2007
## [1] 2455
## [1] 2903
## [1] 3351
```



```

testData = read.csv("log1000.csv")
testData = testData%>%mutate(cummulativeCheaters=cumsum(numCheaters))

testData= testData%>%mutate(time=1:2001)

# ABM log=1000; data only
viz_removalLoggers=testData[1:700,]%>%ggplot(aes(x=time))+
  labs(x="Period",y= axlab_perc )+
  geom_path(aes(y=TotalBM/TotalBM[1]*100, color = "Biomass"),size=line_size)+
  geom_path(aes(y = numLoggers/numLoggers[1]*100, color = "Loggers remaining"),size=line_size)+
  geom_path(aes(y = cummulativeCheaters/numLoggers[1]*100, color = "Loggers removed"),size=line_size)+
  setGraphs+
  theme(legend.title = element_blank())+
  colorDiscrete
ggsave("viz_removalLoggers.pdf", viz_removalLoggers, width=5.5, height=2.5, units="in")

b=data.frame(x=c(1,2,3, 1,2,3), y=as.factor(c(1,1,1,2,2,2)))
testv = b%>%ggplot()+geom_path(aes(x=x, y=1:6, linetype=y))+
  scale_linetype(name = element_blank(), labels = c("ABM", "Lotka-Volterra equations"))+
  setGraphs+
  theme(legend.position = "bottom")
linetype_legend = g_legend(testv)

# changed ABM to make it fit LV equations
viz_abmFitLV=lvTest1%>%slice(1:200)%>%ggplot(aes(x=time))+
  geom_path(aes(x= 1:200, y=TotalBM/TotalBM[1]*100, color="Biomass"), size=line_size)+
  geom_path(aes(x=1:200, y = numLoggers/numLoggers[1]*100, color="Loggers"), size=line_size)+
  geom_path(data=fit_model_to_LV, aes(y=X/xmax*100), color="#66c2a5",size=line_size, linetype="dotted")+
  geom_path(data=fit_model_to_LV, aes(y=Y/ymax*100), color="#fc8d62", size=line_size, linetype="dotted")+
  labs(x="", y="Percent")+
  ylim(0,150)+
  setGraphs+
  theme(legend.title = element_blank(), legend.position = "bottom")+
  #scale_x_continuous(minor_breaks = seq(0, 200, by=50), limits = c(0,200))+
  scale_x_continuous(breaks = seq(0, 200, by=100), minor_breaks = seq(0, 200, by=50), limits = c(0,200))+
  colorDiscrete

lv_legend=g_legend(viz_abmFitLV)

```

```
## Warning: Removed 43 rows containing missing values (geom_path).
```

```
## Warning: Removed 3227 rows containing missing values (geom_path).
```

```
## Warning: Removed 2650 rows containing missing values (geom_path).
```

```
viz_abmFitLV=viz_abmFitLV+theme(legend.position = "none")

# compared actual ABM to LV equations
viz_abmCompareLV=compare_model_to_LV%>%ggplot(aes(x=time))+
  geom_path(aes(y=X/xmax*100), color="#66c2a5", size=line_size, linetype="dotted")+
  geom_path(aes(y=Y/ymax*100), color="#fc8d62", size=line_size, linetype="dotted")+
  geom_path(data = logEnfBM, aes(x= 1:700, y=bm.8.1/bm.8.1[1]*100, color = "Biomass"), s
size=line_size)+
  geom_path(data=logEnfLog, aes(x=1:700, y = log.8.1/log.8.1[1]*100, color = "Loggers"),
size=line_size)+
  ylim(0,150)+xlim(0,450)+
  labs(x="", y=element_blank())+
  setGraphs+
  theme(legend.position = "none")+
  colorDiscrete

viz_abm_lv=grid.arrange(arrangeGrob(viz_abmFitLV+ggtitle("a")),viz_abmCompareLV+ggtitle(
"b")), ncol=2, widths = c(1.2,2)), arrangeGrob(lv_legend, linetype_legend, ncol=2), nrow
=2, heights=c(9,1))
```

```
## Warning: Removed 43 rows containing missing values (geom_path).
```

```
## Warning: Removed 3227 rows containing missing values (geom_path).
```

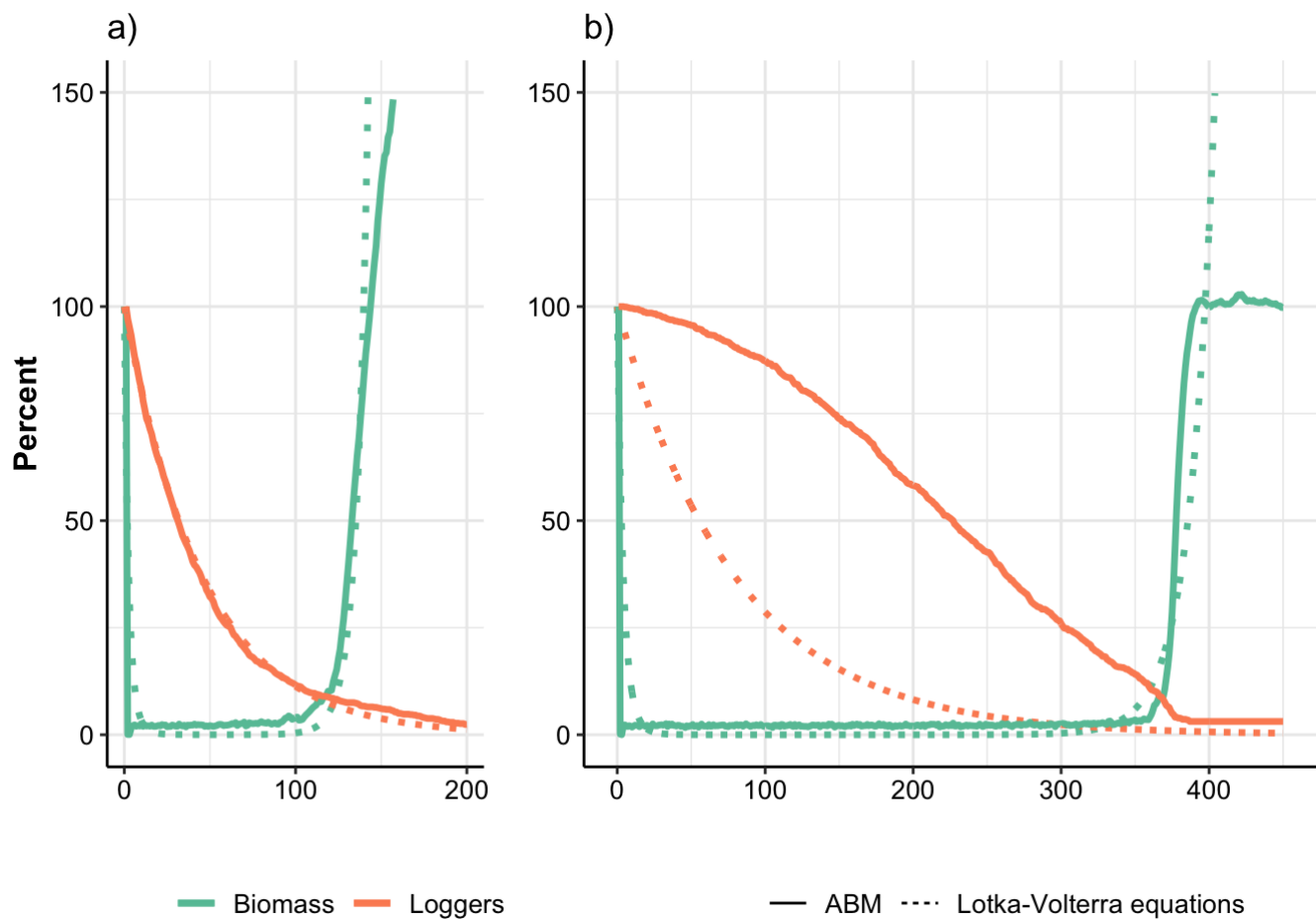
```
## Warning: Removed 2650 rows containing missing values (geom_path).
```

```
## Warning: Removed 611 rows containing missing values (geom_path).
```

```
## Warning: Removed 150 rows containing missing values (geom_path).
```

```
## Warning: Removed 250 rows containing missing values (geom_path).
```

```
## Warning: Removed 250 rows containing missing values (geom_path).
```



```

viz_abm_lv=annotate_figure(viz_abm_lv,
                           bottom = text_grob(expression(bold(Time)), vjust = -3.6, size = axlab_s
                           ize),
                           )

ggsave("viz_abm_lv.pdf", viz_abm_lv, width = 5.75, height = 2.7, units="in")

```

My model

using this - distribution of cheaters

```

setwd("~/Documents/WoosterStuff/fall2019/IS_organized/analysis/allData_IS")
load("cheatProb_distribution.RData")

makeLegend<-cheatProb_distribution%>%filter(variable=="mon.1"|variable=="Sanc.5"|variable=="mon.9")%>%ggplot()+
  geom_histogram(aes(x=cheaters, fill=variable), position="identity", binwidth = 0.05, alpha=0.5)+
  labs(x="Probability of cheating", y=axlab_density)+
  xlim(-0.1,1)+
  scale_fill_manual(name = element_blank(), breaks=c("mon.9", "Sanc.5", "mon.1"), labels = c(expression(paste(italic(monitored-level), " = 0.5", " ", italic(sanction-level), " = 0.5")), expression(paste(italic(monitored-level), " = 0.1", " ", italic(sanction-level), " = 0.5")),expression(paste(italic(monitored-level), " = 0.5", " ", italic(sanction-level), " = 0.1"))), values=c(color3, color1, color2))+
  #theme(legend.position="bottom")+
  guides(fill = guide_legend(title.position="top", title.hjust = 0.5))

monSanc_legend= g_legend(makeLegend)

```

```
## Warning: Removed 6 rows containing missing values (geom_bar).
```

```

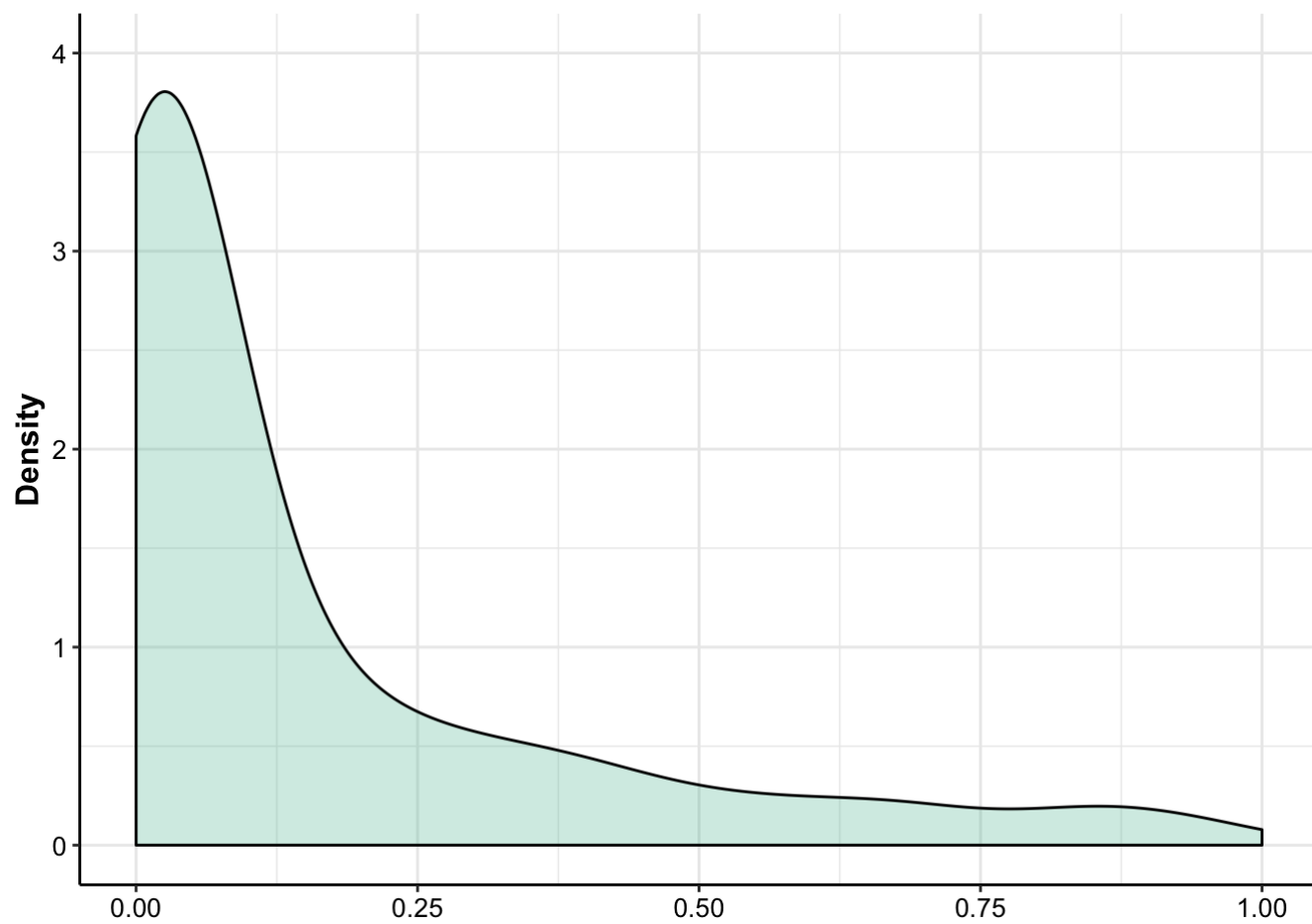
ax_pc = expression(bolditalic(prob-cheat))
v_monLow=cheatProb_distribution%>%filter(variable=="mon.1")%>%ggplot()+
  geom_density(aes(x=cheaters, fill=variable), adjust = 2, fill=color2, alpha=0.3)+
  labs(x=element_blank(), y=element_blank())+
  xlim(0,1)+
  ylim(0,4)+
  setGraphs##+
  #theme(legend.position='none')

v_mid=cheatProb_distribution%>%filter(variable=="Sanc.5")%>%ggplot()+
  geom_density(aes(x=cheaters, fill=variable), adjust = 2, fill=color1, alpha=0.3)+
  labs(x=element_blank(), y=axlab_density)+
  xlim(0,1)+
  ylim(0,4)+
  setGraphs##+
  #theme(legend.position='none')

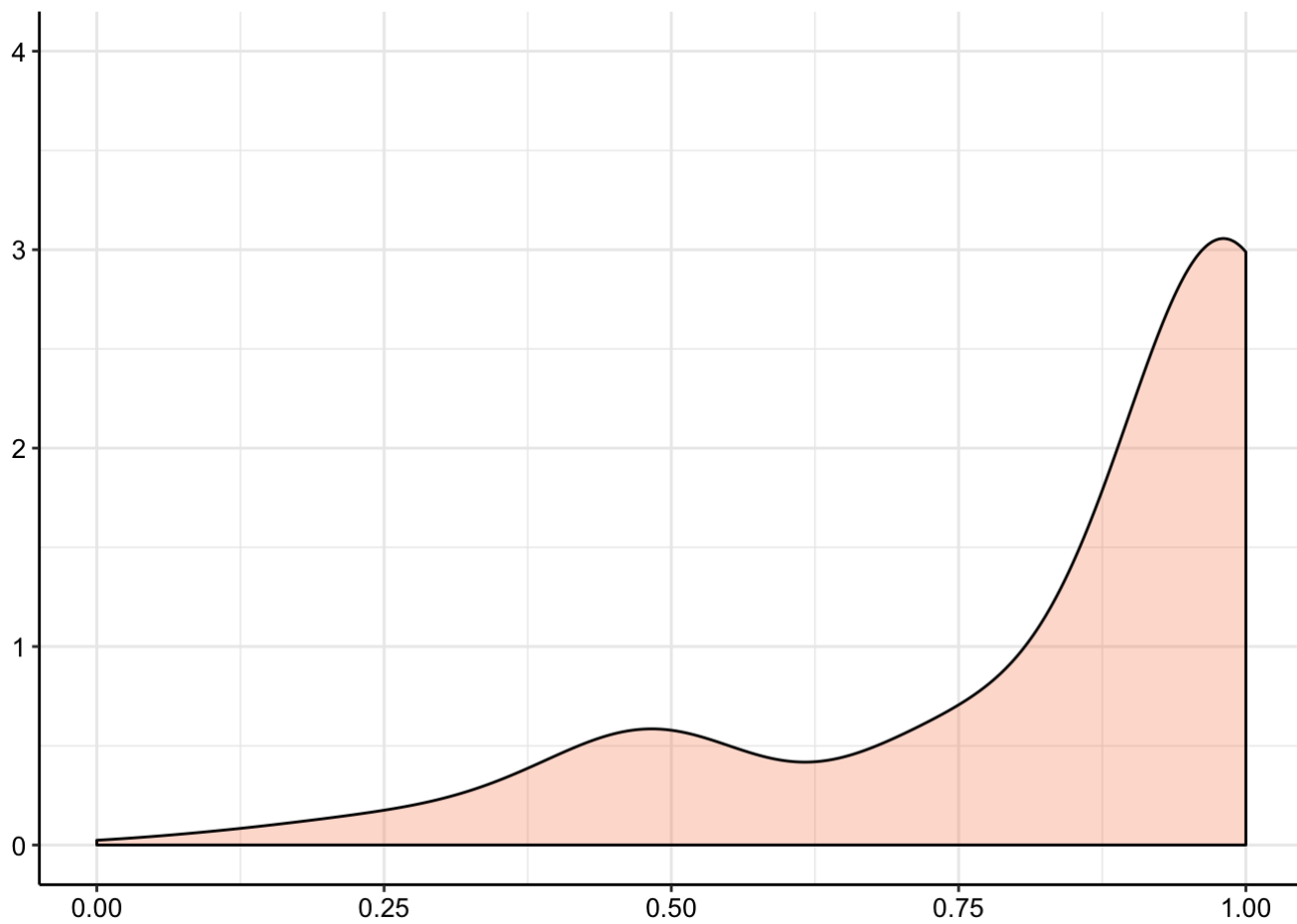
v_sancLow=cheatProb_distribution%>%filter(variable=="Sanc.1")%>%ggplot()+
  geom_density(aes(x=cheaters, fill=variable), adjust = 2, fill=color3, alpha=0.3)+
  labs(x=element_blank(), y=element_blank())+
  xlim(0,1)+
  ylim(0,4)+
  #scale_fill_discrete(name = "monitored-level", labels = c("0.1", "0.5", "0.9"))+
  setGraphs+
  theme(legend.position='none')

v_mid

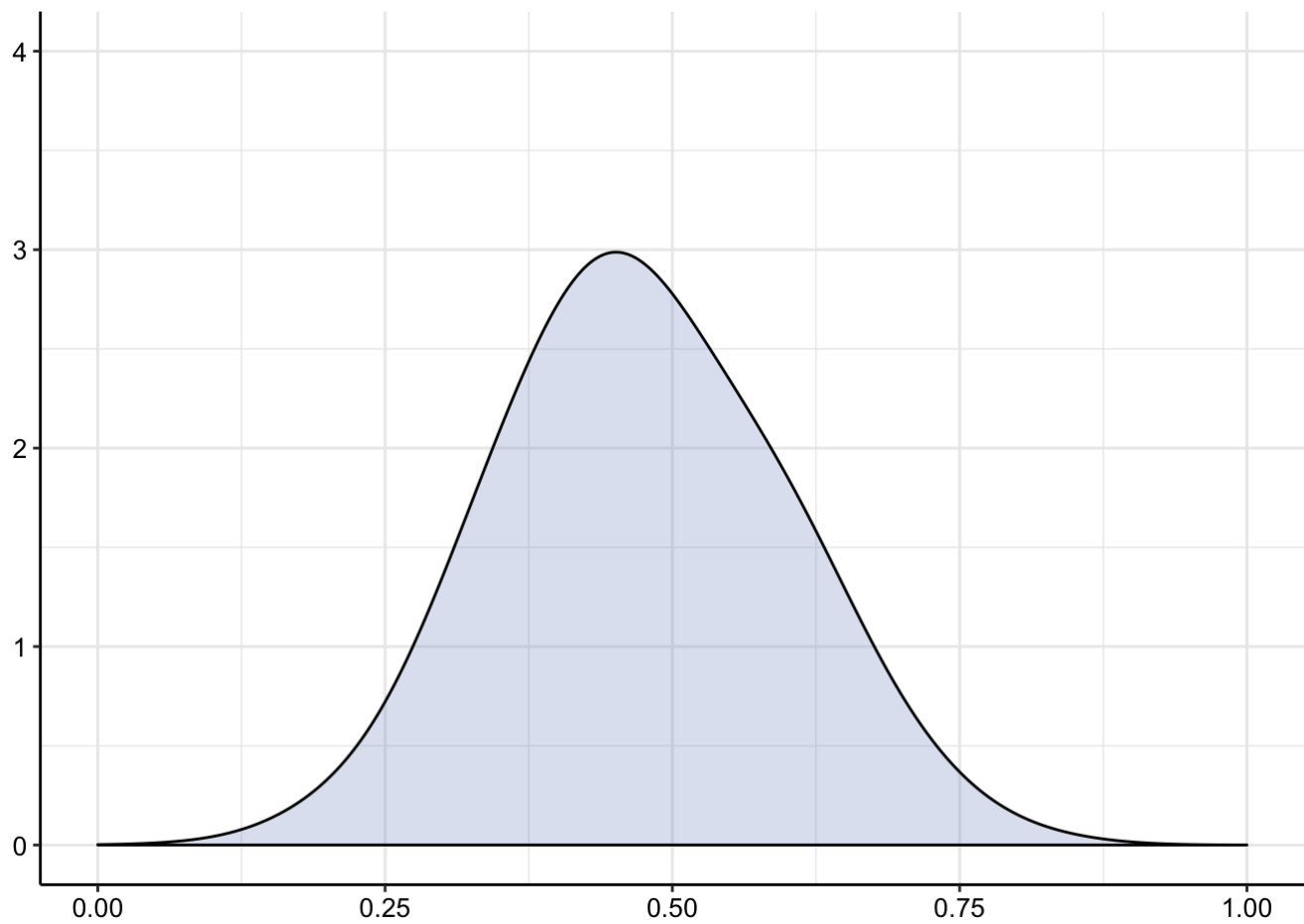
```



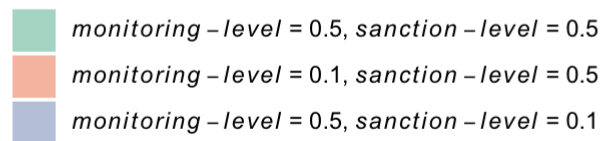
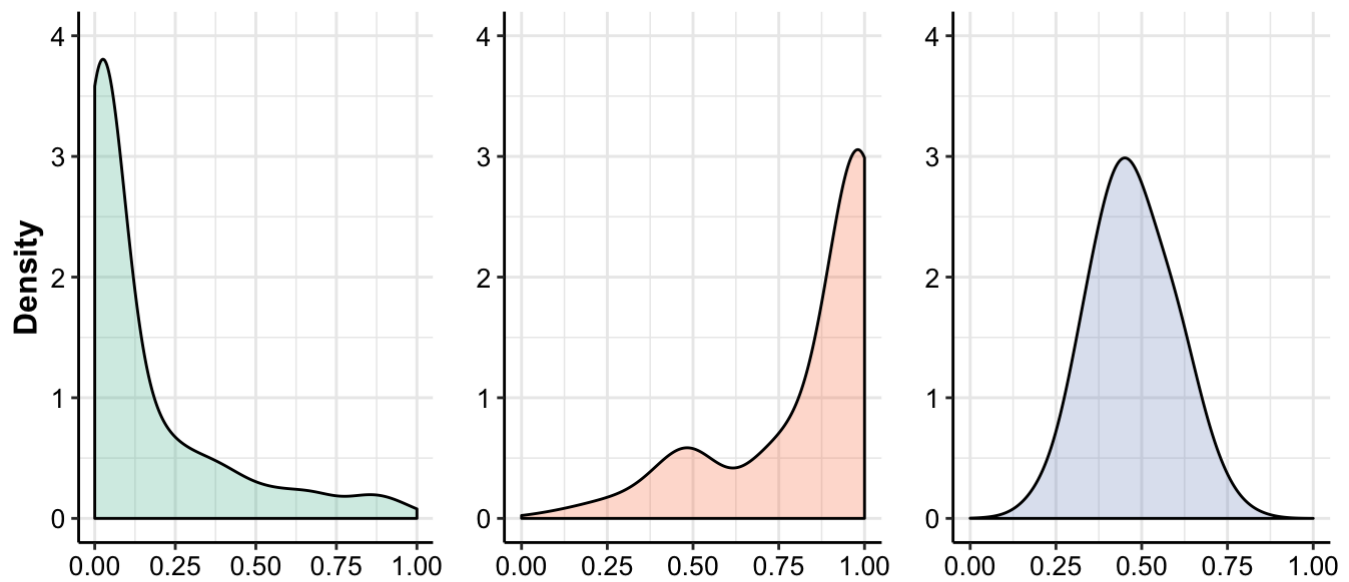
`v_monLow`



v_{sancLow}

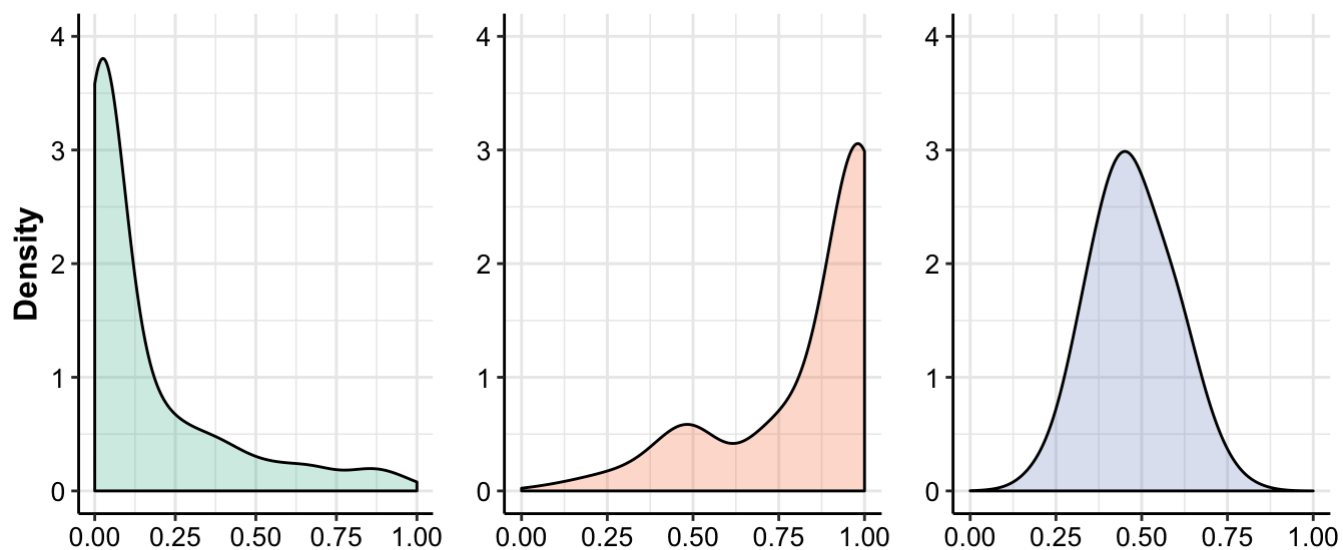


```
viz_monSanc = grid.arrange(arrangeGrob(v_mid, v_monLow, v_sancLow, ncol=3), monSanc_legend, nrow=2, heights=c(7,4))
```



```
viz_monSanc=annotate_figure(viz_monSanc,
                             bottom = text_grob(expression(bolditalic(prob-cheat)), vjust = -8, size
= axlab_size),
                             )

viz_monSanc
```

prob - cheat

- monitoring - level = 0.5, sanction - level = 0.5
- monitoring - level = 0.1, sanction - level = 0.5
- monitoring - level = 0.5, sanction - level = 0.1

```
ggsave("viz_monSanc.pdf", viz_monSanc, width=5.75, height=3.5, units="in")
```

```
cheatProb_distribution%>%filter(variable=="Sanc.9")%>%group_by(cheaters<0.01)%>%summarize(n())
```

```
## # A tibble: 2 x 2
##   `cheaters < 0.01` `n()`
##   <lgl>             <int>
## 1 FALSE              17
## 2 TRUE              983
```

```
cheatProb_distribution%>%filter(variable=="mon.9")%>%group_by(cheaters<0.01)%>%summarize(n())
```

```
## # A tibble: 2 x 2
##   `cheaters < 0.01` `n()`
##   <lgl>             <int>
## 1 FALSE              30
## 2 TRUE             970
```

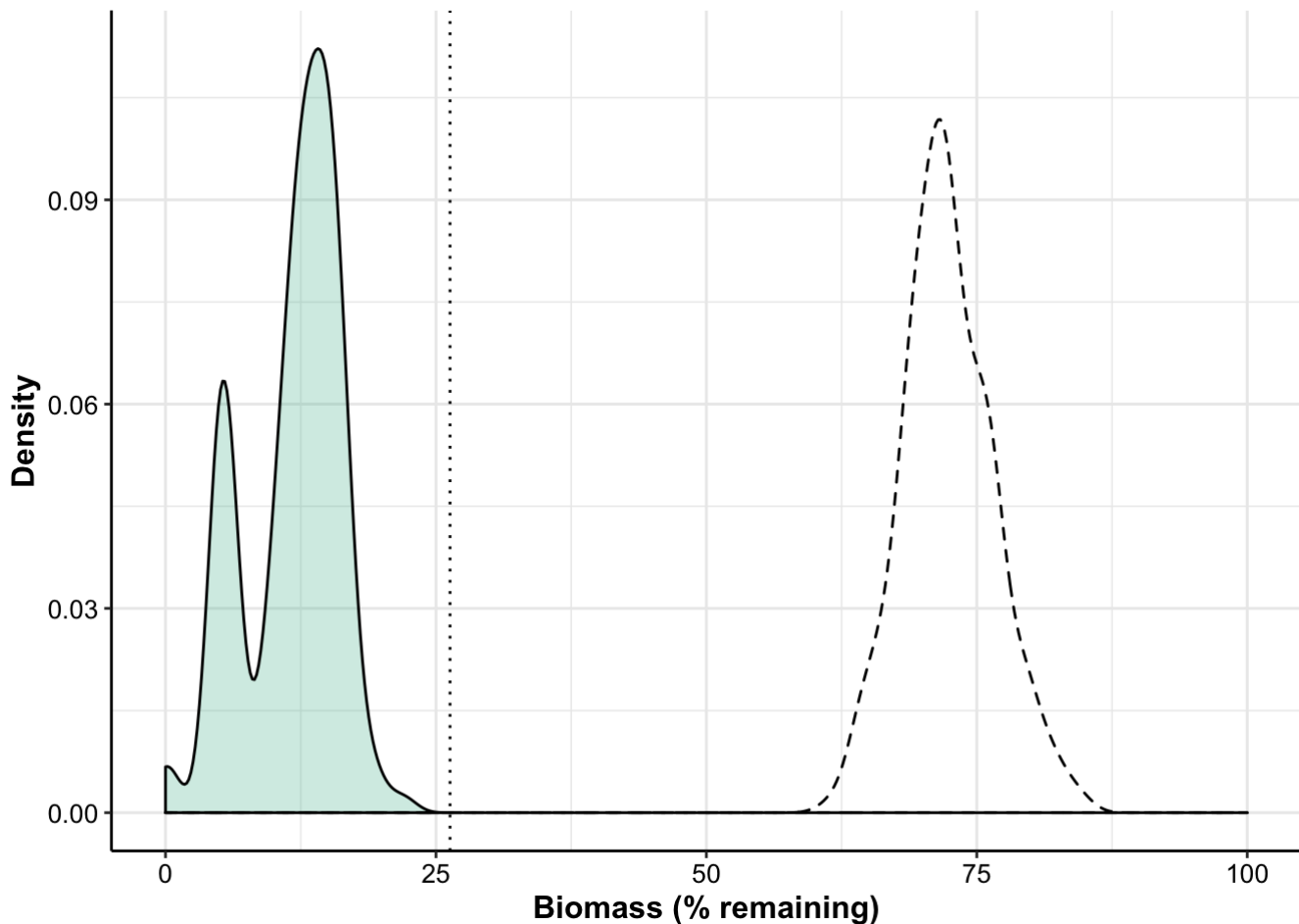
base my model distribution

```

viz_baseMymod=baseData%>%ggplot(aes(x=TotalBM*100, fill="A"))+
  geom_density(alpha=.3)+
  geom_density(data=baseData_vallino, aes(x=TotalBM*100, fill=NULL), linetype="dashed")+
  xlim(0,100)+
  setGraphs+
  theme(legend.position = "none")+
  fillDiscrete+
  labs(x=axlab_bm, y = "Density")+
  geom_vline(xintercept = 26.3, linetype="dotted")
#annotate("text", x=26.3, y=0.12, label="26.3%")
#geom_label(aes(x = 26.3, y = 0.115, label = "26.3"), fill = "white", label.size = 0)

viz_baseMymod

```



```

viz_baseMymod=annotate_figure(viz_baseMymod,
  top = text_grob("26.3%", vjust = 1, hjust=1.5, size = 10),
)

ggsave("viz_baseMymod.pdf", viz_baseMymod, width=4, height=2.5, units="in")

mean(baseData$TotalBM)

```

```
## [1] 0.1170022
```

```
findTrough(baseData$TotalBM, 0.05, 0.15)
```

```
## [1] 0.08105281
```

relationship between K and BM

```
numberJumps<-findKJumps(dataBase, seq(1,6, by=1), 2000, 50)
```

```
## [1] 200 300
```

```
whenFinalJump<-findFinalKJumps(dataBase, seq(1,6, by=1), 2000, 50)

numberJumps2<-numberJumps%>%gather(key="variableValue", value=variableOfInterest)%>%separate(col = variableValue, into = c("type", "variableValue"))

jumpDF<-data.frame(jumps=numberJumps2$variableOfInterest, K=baseData$K, TotalBM=baseData$TotalBM, variableValue=baseData$variableValue, cheaters=baseData$numCheaters)

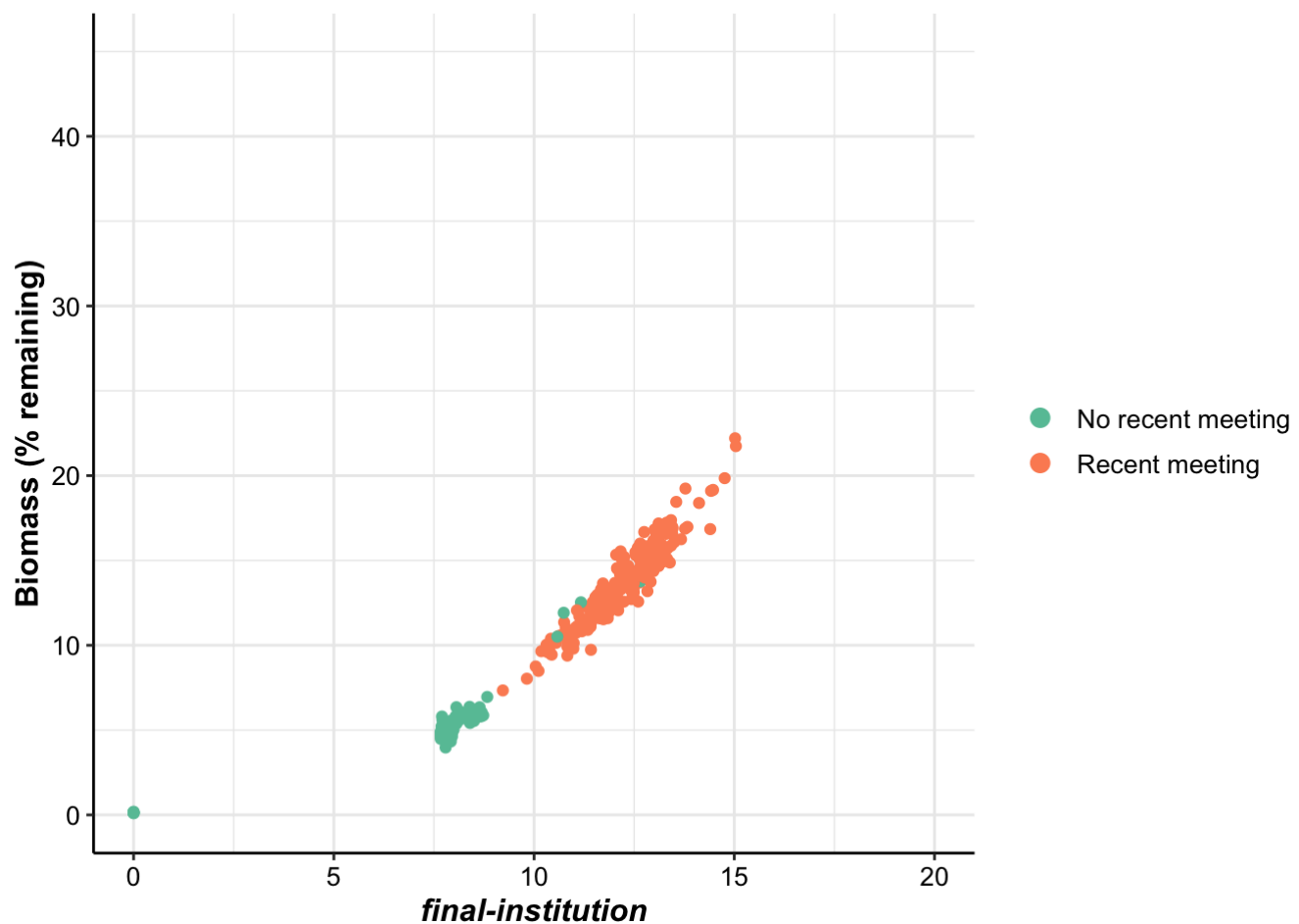
whenFinalJump2<-whenFinalJump%>%gather(key="variableValue", value=variableOfInterest)%>%separate(col = variableValue, into = c("type", "variableValue"))
baseJumpData<-data.frame(numJumps = numberJumps2$variableOfInterest, finalJump=whenFinalJump2$variableOfInterest, K=baseData$K, TotalBM=baseData$TotalBM, variableValue=baseData$variableValue, cheaters=baseData$numCheaters)

#####
trough= findTrough(baseJumpData$finalJump, 20, 50)
baseJumpData<- baseJumpData%>%mutate(recentMeeting = finalJump>trough)
#####

allData_plotK_BM<-allData%>%mutate(logger60= ((param=="log") & (variableValue==60)))

kBM_base_plot<-baseJumpData%>%ggplot(aes(K, TotalBM*100))+
  geom_point(aes(color=recentMeeting))+
  xlim(0,20)+ ylim(0,45)+
  setGraphs+
  scale_color_manual(labels = c("No recent meeting", "Recent meeting"), values=c(color1, color2))+
  theme(legend.title = element_blank())+
  labs(y=axlab_bm, x=ax_kf)+
  guides(color = guide_legend(override.aes = list(size = 3)))

kBM_base_plot
```



```
# transparent?
kBM_allData_plot<-allData_plotK_BM%>%ggplot(aes(K, TotalBM*100, color=logger60))+
  geom_point()+
  xlim(0,20)+ ylim(0,45)+
  setGraphs +
  scale_color_manual(breaks=c(TRUE, FALSE), labels = c("60 Loggers", "All other parameter sets"), values=c(color1, color2))+
  theme(legend.title = element_blank())+
  labs(y=axlab_bm, x=ax_kf)+
  guides(color = guide_legend(override.aes = list(size = 3)))

#grid.arrange(kBM_allData_plot2, kBM_allData_plot)
ggsave("viz_kBM_base.pdf", kBM_base_plot, width=4.5, height=2.5, units="in")
ggsave("viz_kBM_allData.pdf", kBM_allData_plot, width=5, height=2.5, units="in")

cor(allData$TotalBM, allData$K)
```

```
## [1] 0.8523361
```

```
cor(baseData$TotalBM, baseData$K)
```

```
## [1] 0.9418734
```

```
cor((allData_plotK_BM%>%filter(!logger60))$TotalBM, (allData_plotK_BM%>%filter(!logger60))$K)
```

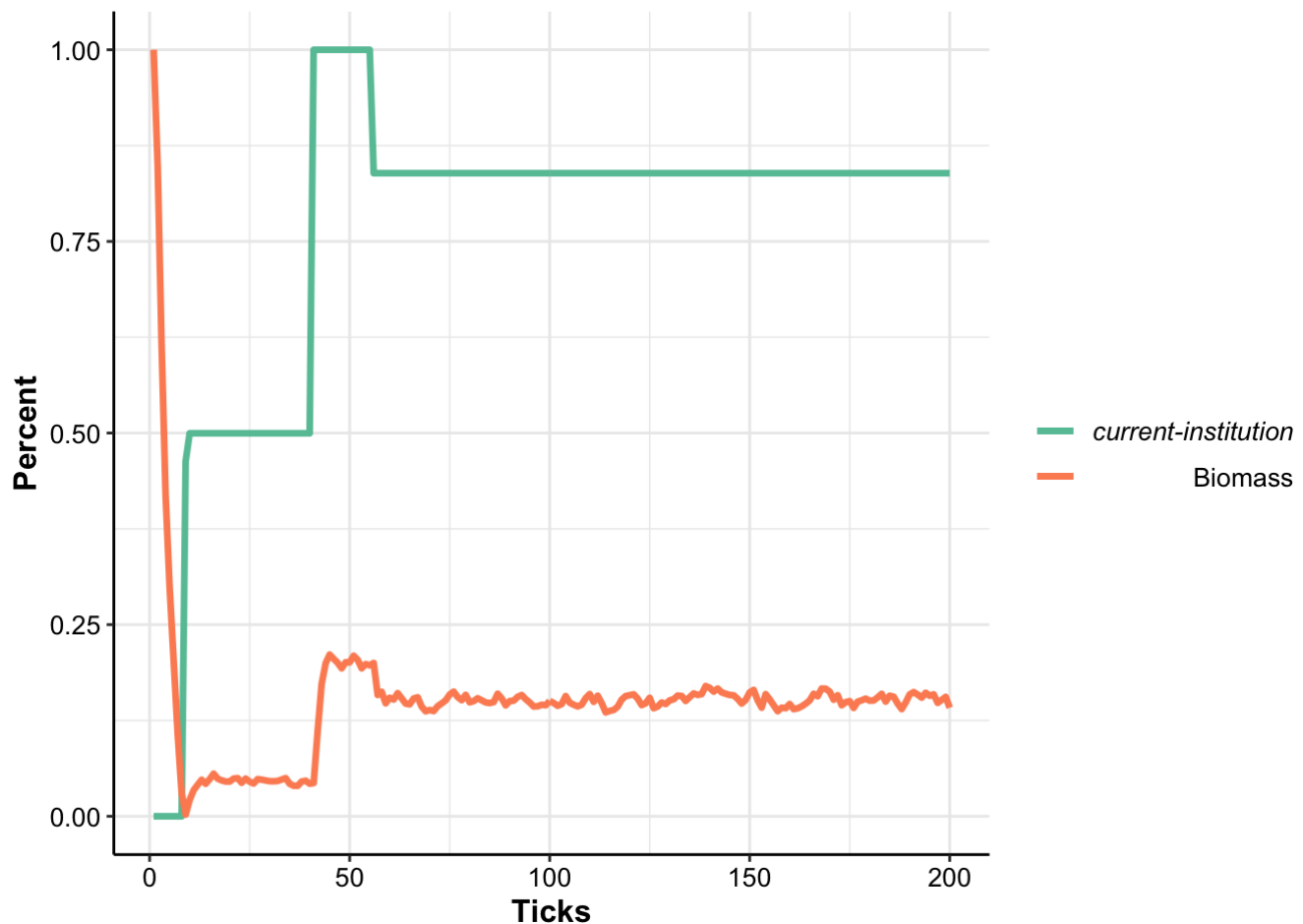
```
## [1] 0.9443373
```

jump relationship between K and BM

```
maxK1 = max(dataBase[[1]][[1]]$K)
v1=dataBase[[1]][[1]]%>%ggplot(aes(x=1:200))+
  geom_path(aes(y=K/maxK1, color=color1), size=line_size)+
  geom_path(aes(y=TotalBM/TotalBM[1], color=color2), size=line_size)+
  setGraphs+
  colorDiscrete+
  scale_colour_manual(values= c(color1, color2), name = "",
                      breaks=c(color1, color2),
                      labels=c(expression(italic("current-institution")), "Biomass"))+
  labs(x="Ticks", y="Percent")
```

```
## Scale for 'colour' is already present. Adding another scale for
## 'colour', which will replace the existing scale.
```

```
v1
```



```
v2=dataBase[[1]][[1]]%>%ggplot(aes(x=1:200))+  
  geom_path(aes(y=K, color=color1), size=line_size)+  
  setGraphs+  
  colorDiscrete+  
  scale_colour_manual(values= c("black"), name = "")+  
  labs(x="", y=ax_k)+  
  theme(legend.position="none")+  
  geom_vline(xintercept = 8, color="dimgrey", linetype="dashed")+  
  geom_vline(xintercept = 59, color="dimgrey", linetype="dashed")
```

```
## Scale for 'colour' is already present. Adding another scale for  
## 'colour', which will replace the existing scale.
```

```
v3=dataBase[[1]][[1]]%>%ggplot(aes(x=1:200))+  
  geom_path(aes(y=TotalBM/TotalBM[1], color=color2), size=line_size)+  
  setGraphs+  
  colorDiscrete+  
  scale_colour_manual(values= c("black"), name = "", labels=c("Biomass"))+  
  labs(x="Period", y=axlab_bm)+  
  theme(legend.position="none")+  
  geom_vline(xintercept = 8, color="dimgrey", linetype="dashed")+  
  geom_vline(xintercept = 59, color="dimgrey", linetype="dashed")
```

```
## Scale for 'colour' is already present. Adding another scale for  
## 'colour', which will replace the existing scale.
```

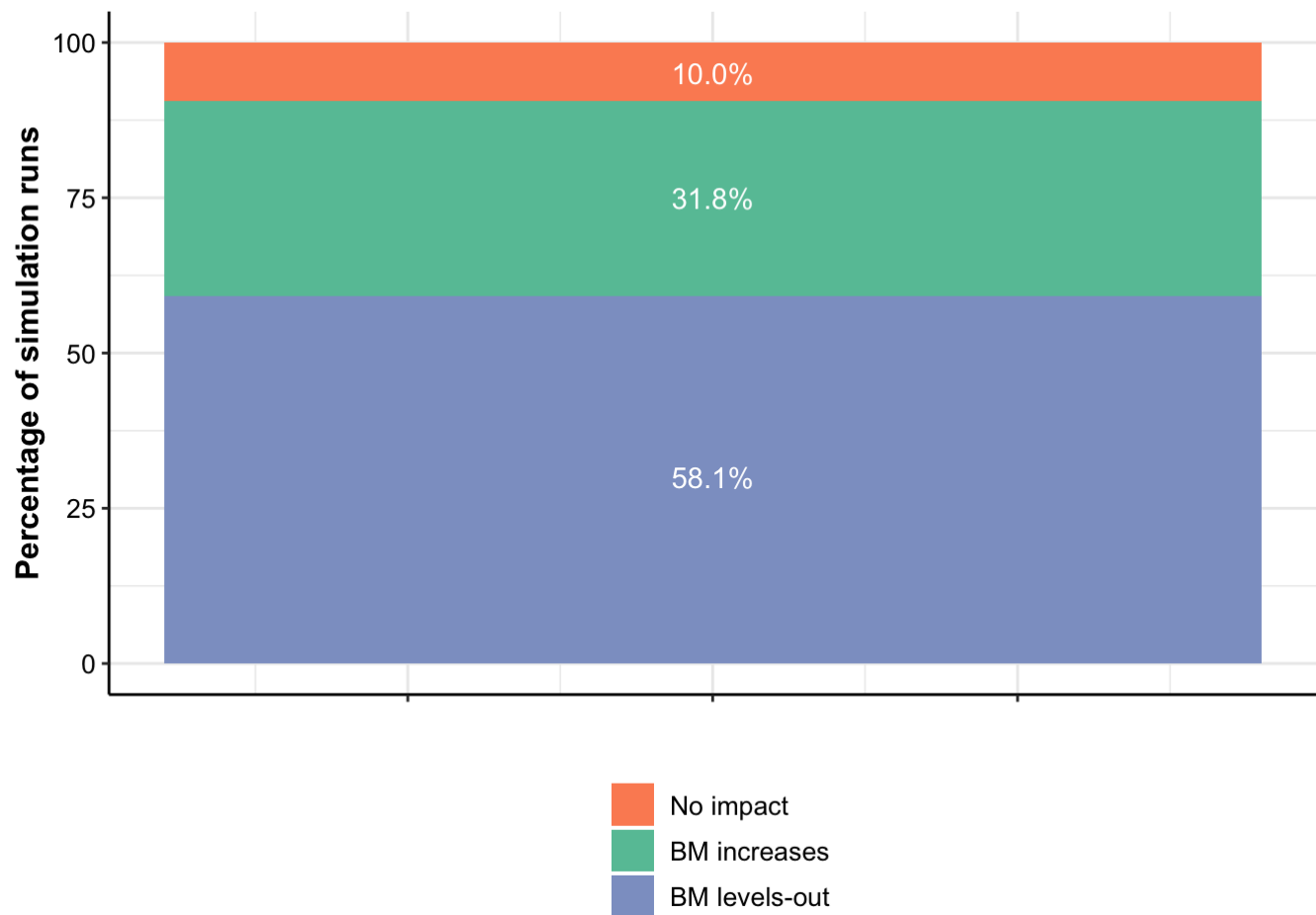
```

both_jump = jumps%>%filter(yep!=0)%>%group_by(yep)%>%summarise(prop=n())
both_jump = both_jump%>%mutate(prop=prop/sum(prop)*100, yep=as.factor(yep))%>%arrange(prop)
both_jump$yep <- relevel(both_jump$yep, "BM remains constant")

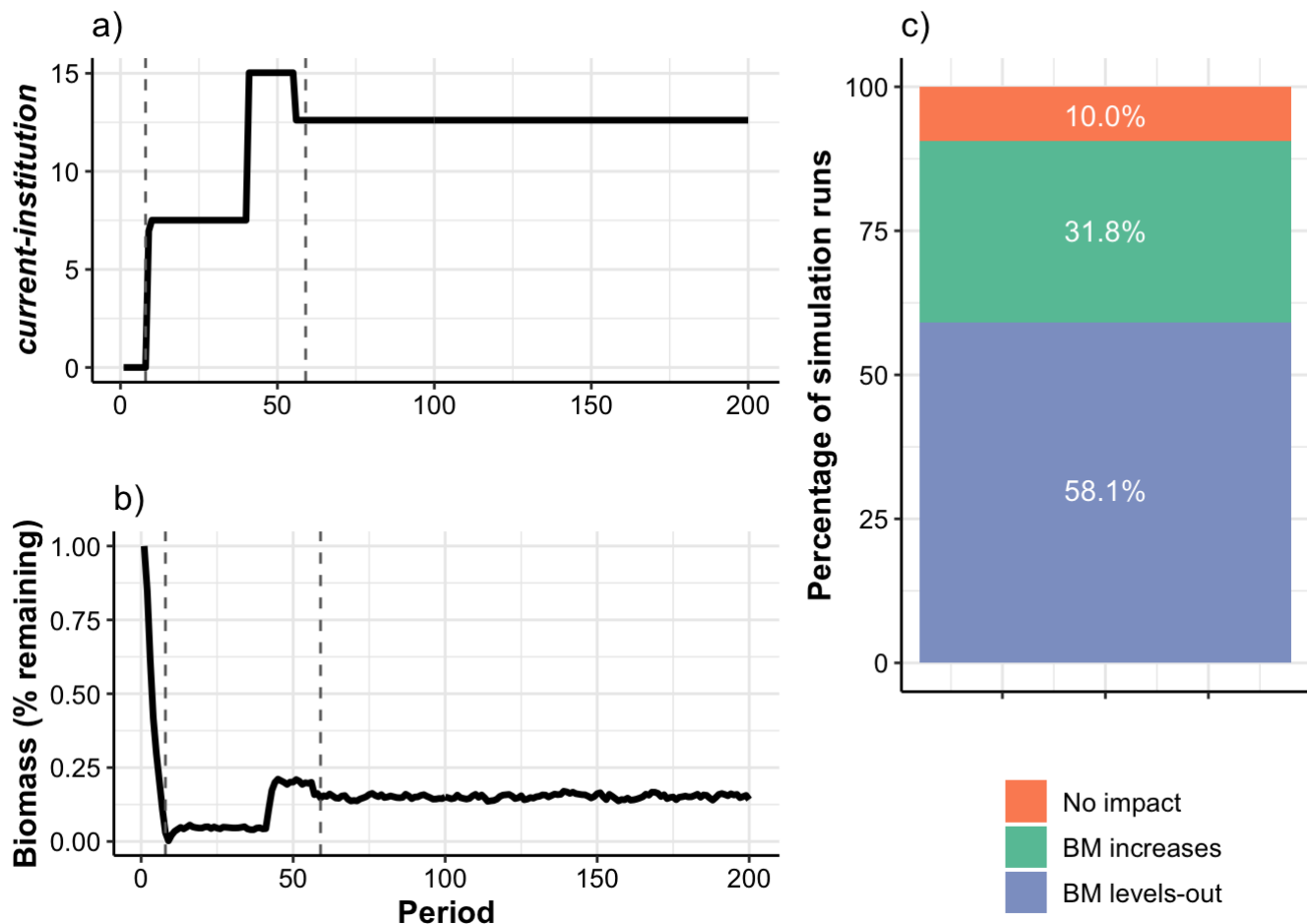
v4=both_jump%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+
  labs(fill= "", x="", y="Percentage of simulation runs")+
  theme(axis.text.x = element_blank()) +
  annotate("text", x = 1, y = 95, label = "10.0%", color="white")+
  annotate("text", x = 1, y = 75, label = "31.8%", color="white")+
  annotate("text", x = 1, y = 30, label = "58.1%", color="white")+
  theme(legend.position="bottom")+
  #theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
  scale_fill_manual(labels=c("No impact", "BM increases", "BM levels-out"), values=c(color2, color1, color3))
mylegend<-g_legend(v4)
v4=both_jump%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+
  labs(fill= "", x="", y="Percentage of simulation runs")+
  theme(axis.text.x = element_blank()) +
  annotate("text", x = 1, y = 95, label = "10.0%", color="white")+
  annotate("text", x = 1, y = 75, label = "31.8%", color="white")+
  annotate("text", x = 1, y = 30, label = "58.1%", color="white")+
  theme(legend.position="none")+
  #theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
  scale_fill_manual(labels=c("No impact", "BM increases", "BM levels-out"), values=c(color2, color1, color3))

v4=both_jump%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+
  labs(fill= "", x="", y="Percentage of simulation runs")+
  theme(axis.text.x = element_blank()) +
  annotate("text", x = 1, y = 95, label = "10.0%", color="white")+
  annotate("text", x = 1, y = 75, label = "31.8%", color="white")+
  annotate("text", x = 1, y = 30, label = "58.1%", color="white")+
  theme(legend.position="bottom")+
  #theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
  scale_fill_manual(labels=c("No impact", "BM increases", "BM levels-out"), values=c(color2, color1, color3))+
  guides(fill=guide_legend(nrow=3))
v4

```



```
viz_jumps=grid.arrange(arrangeGrob(arrangeGrob(arrangeGrob(v2+ggtitle("a")),v3+ggtitle("b")), ncol=1, nrow=2)), v4+ggtitle("c")), ncol=2, widths=c(3,2))
```

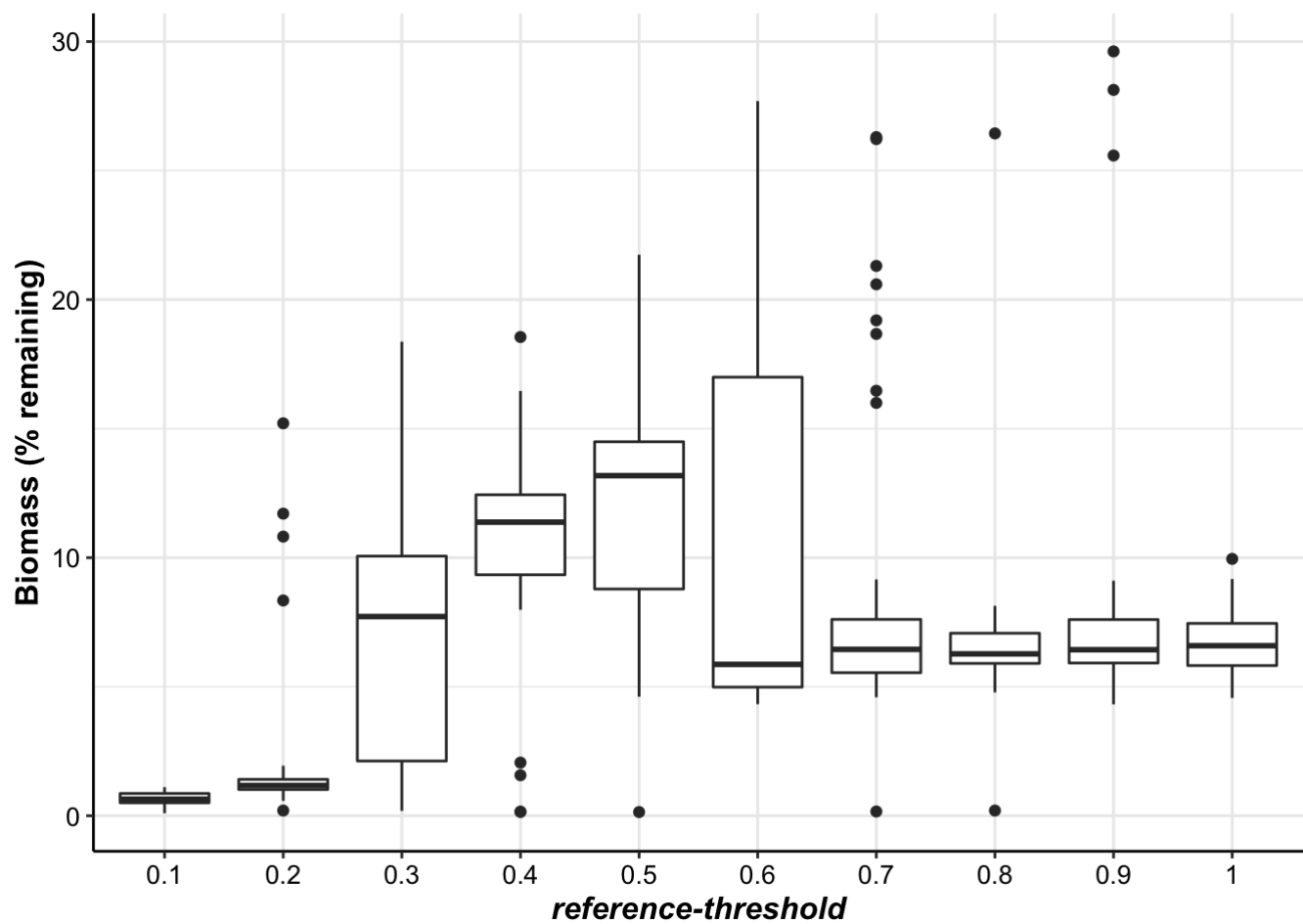
```
#viz_jumps=grid.arrange(arrangeGrob(v2,v3, ncol=1, nrow=2),
#      arrangeGrob(v4, mylegend, ncol=1, nrow=2, heights=c(9,1)), widths=c(3,2))
#viz_jumps=grid.arrange(arrangeGrob(arrangeGrob(arrangeGrob(v2+ggtitle("a"),v3+ggtitle
("b")), ncol=1, nrow=2), #v4+ggtitle("c")), ncol=2, widths=c(3,2)), mylegend, nrow=2, he
ights=c(20,1)))

ggsave("viz_jumps.pdf", viz_jumps, width = 5, height = 5, units = "in")
```

rt boxplot

```
rtData2<-rtData%>%mutate(param="rt")%>%addMeans()

#rt_boxplot<-rtData%>%ggplot(aes(x=variableValue, y=TotalBM, fill=eval(parse(text = fill
Variable))))+
# divide variableValue by 10 bc it was multiplied by ten to make data manipulation easie
r
rt_boxplot<-rtData2%>%ggplot(aes(x=as.factor(as.numeric(variableValue)/10), y=TotalBM*10
0))+
  geom_boxplot()+
  fillCont+
  setGraphs+
  labs(y=axlab_bm, x=expression(bolditalic("reference-threshold"))))
rt_boxplot
```



```
ggsave("viz_rtBoxplot.pdf", rt_boxplot, width=4, height=2.5, units="in")
```

calculating jumps

```
x<-maxK(dataRT, seq(1,10, by=1), 2000, 50)

#jumps_test<-findKJumps(dataRT, seq(1,10, by=1), 2000, 50)
numberJumps<-findKJumps(dataRT, seq(1,10, by=1), 2000, 50)
```

```
## [1] 200 500
```

```

whenFinalJump<-findFinalKJumps(dataRT, seq(1,10, by=1), 2000, 50)

numberJumps2<-numberJumps%>%gather(key="variableValue", value=variableOfInterest)%>%separate(col = variableValue, into = c("type", "variableValue"))

jumpDF<-data.frame(maxK=x, jumps=numberJumps2$variableOfInterest, K=rtData$K, TotalBM=rtData$TotalBM, variableValue=rtData$variableValue, cheaters=rtData$numCheaters)

whenFinalJump2<-whenFinalJump%>%gather(key="variableValue", value=variableOfInterest)%>%separate(col = variableValue, into = c("type", "variableValue"))
rtJumpData<-data.frame(maxK=x, numJumps = numberJumps2$variableOfInterest, finalJump=whenFinalJump2$variableOfInterest, K=rtData$K, TotalBM=rtData$TotalBM, variableValue=rtData$variableValue, cheaters=rtData$numCheaters)

rtJumpData<- rtJumpData%>%mutate(recentMeeting = finalJump>20)
rtJumpData<- rtJumpData%>%mutate(jumpCat = ifelse(as.numeric(numJumps)<2,"once", ifelse(as.numeric(numJumps)<3, "twice","3+ times")))%>%mutate(jumpCat=factor(jumpCat))

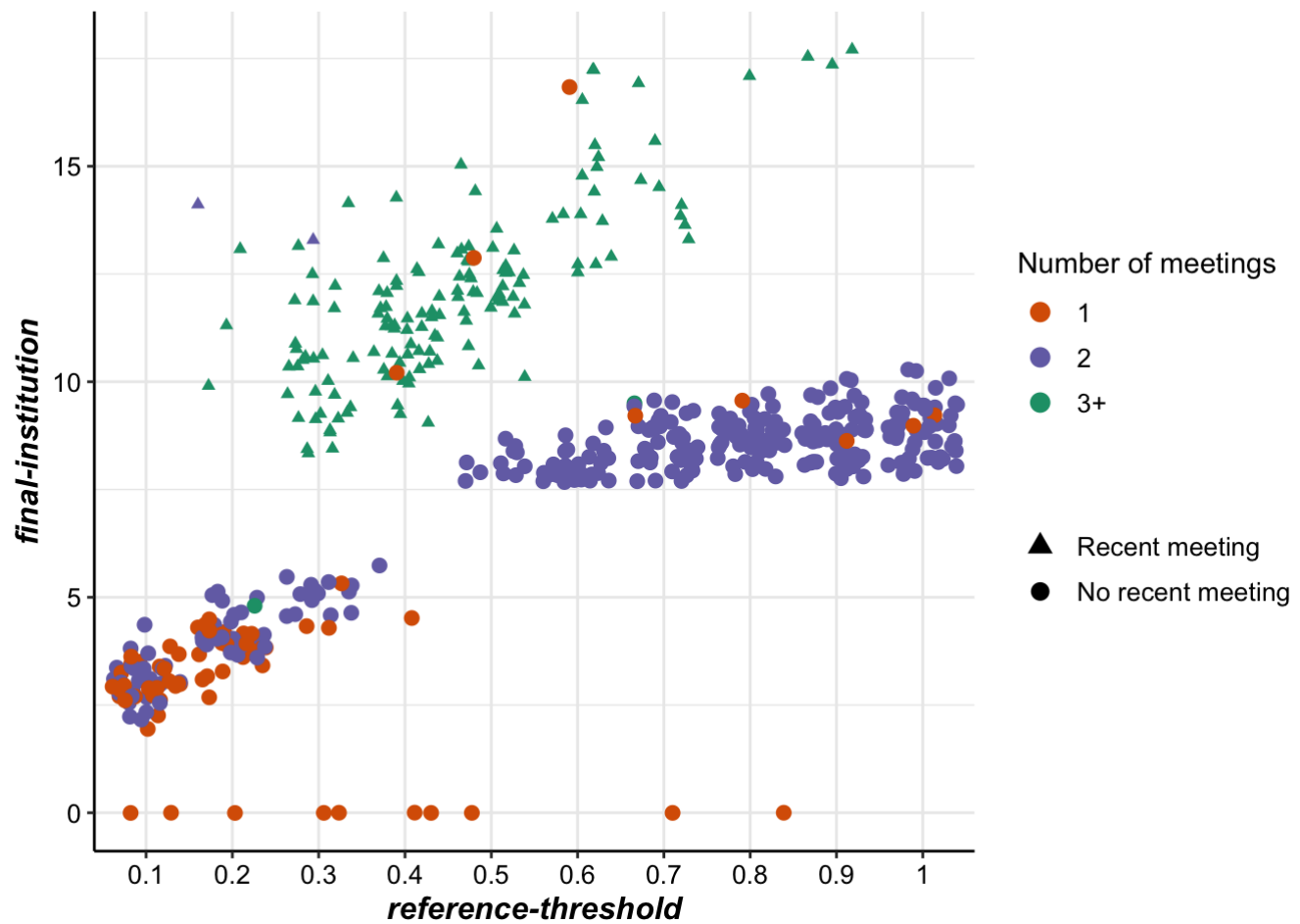
```

rt jumps scatterplot

```

viz_rt_jump=rtJumpData%>%ggplot(aes(as.factor(as.numeric(variableValue)/10), K))+
  geom_point(aes(color=jumpCat, shape=recentMeeting, size=recentMeeting), position="jitter")+
  scale_color_brewer(palette="Dark2",breaks=c("once", "twice", "3+ times"),name="Number of meetings", labels = c("1", "2", "3+"))+
  scale_shape(name="", breaks=c(TRUE, FALSE), labels=c("Recent meeting", "No recent meeting"))+
  scale_size_manual(breaks=c(TRUE, FALSE), values=c(2.5,1.5))+
  labs(x=ax_rt, y=ax_kf)+
  setGraphs+
  guides(color = guide_legend(override.aes = list(size = 3)), shape = guide_legend(override.aes = list(size = 3)), size=FALSE)
viz_rt_jump

```



```
ggsave("viz_rtJump.pdf", viz_rt_jump, width=5.75, height=3, units="in")

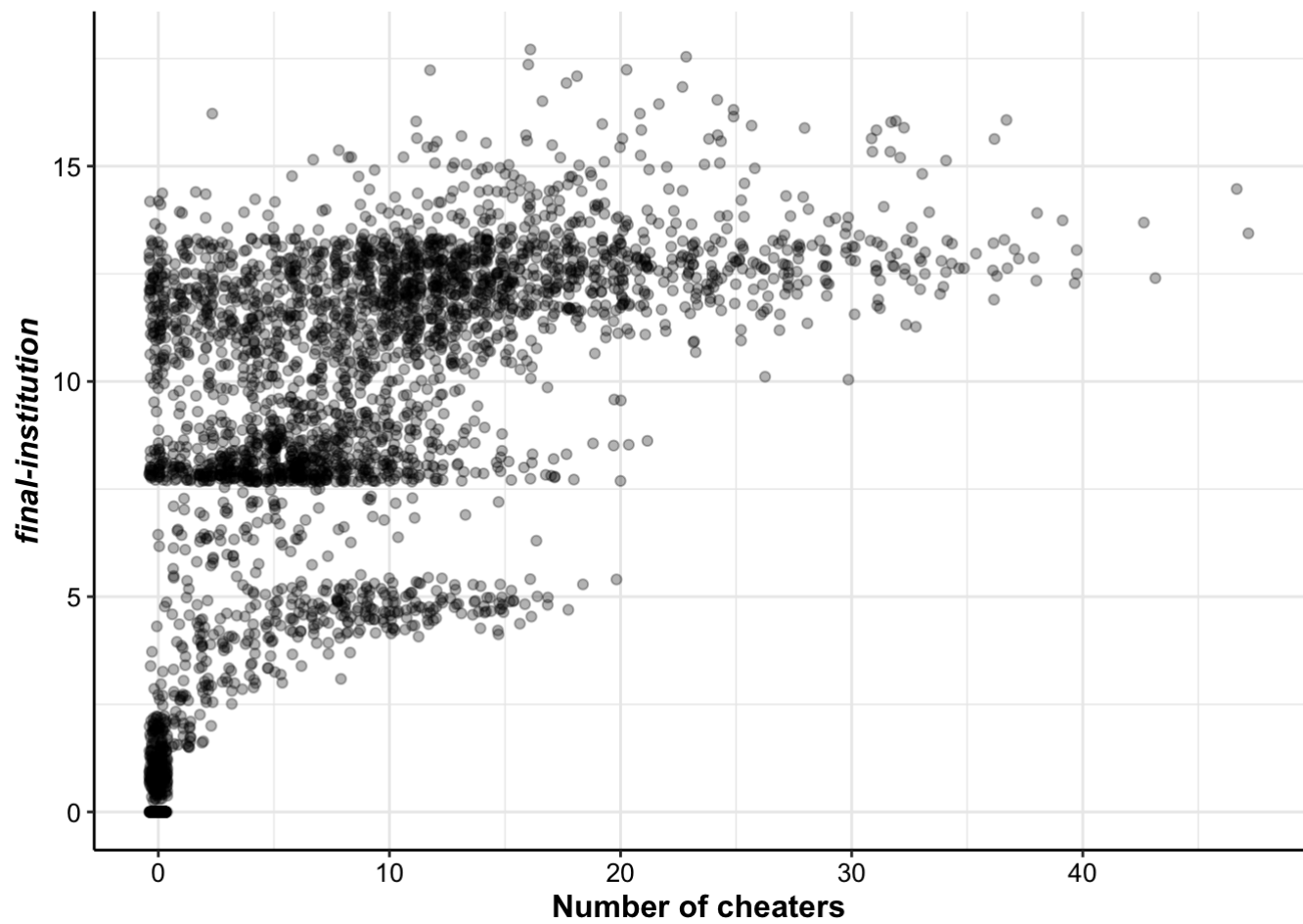
cor(allData$K, allData$numCheaters)
```

```
## [1] 0.5742562
```

relationship between numCheaters and K

```
viz_cheatK_cor=allData%>%ggplot(aes(x=numCheaters, y = K, color))+
  geom_point(position="jitter", alpha=0.3)+
  #geom_smooth(method='glm', se=FALSE, color=color1)+
  labs(x="Number of cheaters", y = ax_kf)+
  setGraphs

viz_cheatK_cor
```



```
ggsave("viz_cheatK_cor.pdf", viz_cheatK_cor, width=4, height=2.5, units="in")
```

Jumps for allData - not using

```

x<-maxK(dataAll, seq(1,68, by=1), 2000, 50)

#jumps_test<-findKJumps(dataRT, seq(1,10, by=1), 2000, 50)
numberJumps<-findKJumps(dataAll, seq(1,68, by=1), 2000, 50)
whenFinalJump<-findFinalKJumps(dataAll, seq(1,68, by=1), 2000, 50)

numberJumps2_all<-numberJumps%>%gather(key="variableValue", value=variableOfInterest)%>%
separate(col = variableValue, into = c("type", "variableValue"))

jumpDF_all<-data.frame(maxK=x, jumps=numberJumps2_all$variableOfInterest, K=allData$K, TotalBM=allData$TotalBM, variableValue=allData$variableValue, cheaters=allData$numCheaters)

whenFinalJump2_all<-whenFinalJump%>%gather(key="variableValue", value=variableOfInterest)%>%separate(col = variableValue, into = c("type", "variableValue"))
allJumpData<-data.frame(maxK=x, numJumps = numberJumps2_all$variableOfInterest, finalJump=whenFinalJump2_all$variableOfInterest, K=allData$K, TotalBM=allData$TotalBM, variableValue=allData$variableValue, cheaters=allData$numCheaters)

allJumpData = allJumpData%>%mutate(recentMeeting = finalJump>100)
allJumpData = allJumpData%>%mutate(jumpCat = ifelse(as.numeric(numJumps)<2,"once", ifelse(as.numeric(numJumps)<3, "twice","3+ times")))%>%mutate(jumpCat=factor(jumpCat))

unique(allJumpData$numJumps)

allJumpData%>%ggplot(aes(x=cheaters, y = round(K,1), color=jumpCat, shape=recentMeeting))+
  geom_count(position="jitter")+
  #geom_smooth(method='glm', se=FALSE, color=color1)+
  labs(x="Number of cheaters", y = ax_kf)+
  setGraphs

head(allJumpData)

```

```

kjumpsAll = findKJumpTimes(dataAll, 1:6, 2000, 50)
bmjumpsAll = findBMJumpTimes(dataAll, 1:6, 2000, 50)

jumpsAll= kjumpsAll%>%gather(key="run", value = "k.jump")
jumps2All= bmjumpsAll%>%gather(key="run", value = "bm.jump")
jumpsAll = jumpsAll%>%mutate(bm.jump=jumps2All$bm.jump, yep=0)

distance = 2
for(i in distance:(dim(jumpsAll)[1])){
  if (jumpsAll$k.jump[i]== 1){
    if(any(jumpsAll$bm.jump[i:(i+distance)]==1)){
      jumpsAll$yep[i] = "BM jumps up"
    }
    else if(all(jumps$bm.jump[(i-distance):i]==-1) & any(jumps$bm.jump[i:(i+distance)]==
0)){
      jumpsAll$yep[i] = "BM stops decreasing"
    }
    else{jumpsAll$yep[i] = "BM remains constant"}
  }
}

```

```

maxK1 = max(dataBase[[1]][[1]]$K)

both_jumpAll = jumpsAll%>%filter(yep!=0)%>%group_by(yep)%>%summarise(prop=n())
both_jumpAll = both_jumpAll%>%mutate(prop=prop/sum(prop)*100, yep=as.factor(yep))%>%arrange(prop)
both_jumpAll$yep <- relevel(both_jumpAll$yep, "BM remains constant")

v4=both_jumpAll%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+
  labs(fill= "", x="", y="Percentage of simulation runs")+
  theme(axis.text.x = element_blank()) +
  annotate("text", x = 1, y = 95, label = "10.0%", color="white")+
  annotate("text", x = 1, y = 75, label = "31.8%", color="white")+
  annotate("text", x = 1, y = 30, label = "58.1%", color="white")+
  theme(legend.position="bottom")+
  #theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
  scale_fill_manual(labels=c("No impact", "BM increases", "BM levels-out"), values=c(color2, color1, color3))
mylegend<-g_legend(v4)
v4=both_jumpAll%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+
  labs(fill= "", x="", y="Percentage of simulation runs")+
  theme(axis.text.x = element_blank()) +
  annotate("text", x = 1, y = 95, label = "10.0%", color="white")+
  annotate("text", x = 1, y = 75, label = "31.8%", color="white")+
  annotate("text", x = 1, y = 30, label = "58.1%", color="white")+
  theme(legend.position="none")+
  #theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
  scale_fill_manual(labels=c("No impact", "BM increases", "BM levels-out"), values=c(color2, color1, color3))

v4=both_jumpAll%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+
  labs(fill= "", x="", y="Percentage of simulation runs")+
  theme(axis.text.x = element_blank()) +
  annotate("text", x = 1, y = 88, label = "26.9%", color="white")+
  annotate("text", x = 1, y = 63, label = "22.8%", color="white")+
  annotate("text", x = 1, y = 25, label = "50.4%", color="white")+
  theme(legend.position="bottom")+
  #theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
  scale_fill_manual(labels=c("No impact", "BM increases", "BM levels-out"), values=c(color2, color1, color3))+
  guides(fill=guide_legend(nrow=3))

# relevel
both_jumpAll$yep <- relevel(both_jumpAll$yep, "BM jumps up")
v5=both_jumpAll%>%ggplot(aes(x=1, y = prop))+
  geom_col(aes(fill=yep))+
  setGraphs+

```



```

labs(fill= "", x="", y="Percentage of simulation runs")+
theme(axis.text.x = element_blank()) +
annotate("text", x = 1, y = 88, label = "22.8%", color="white")+
annotate("text", x = 1, y = 63, label = "26.9%", color="white")+
annotate("text", x = 1, y = 25, label = "50.4%", color="white")+
theme(legend.position="bottom")+
#theme(legend.position = c(0.8, 0.2), legend.direction = "horizontal")+
scale_fill_manual(labels=c("BM increases", "No impact", "BM levels-out"), values=c(col
or1, color2, color3))+
guides(fill=guide_legend(nrow=3))

```

```

both_jumpAll%>%group_by(prop)%>%summarize(n())

```

```

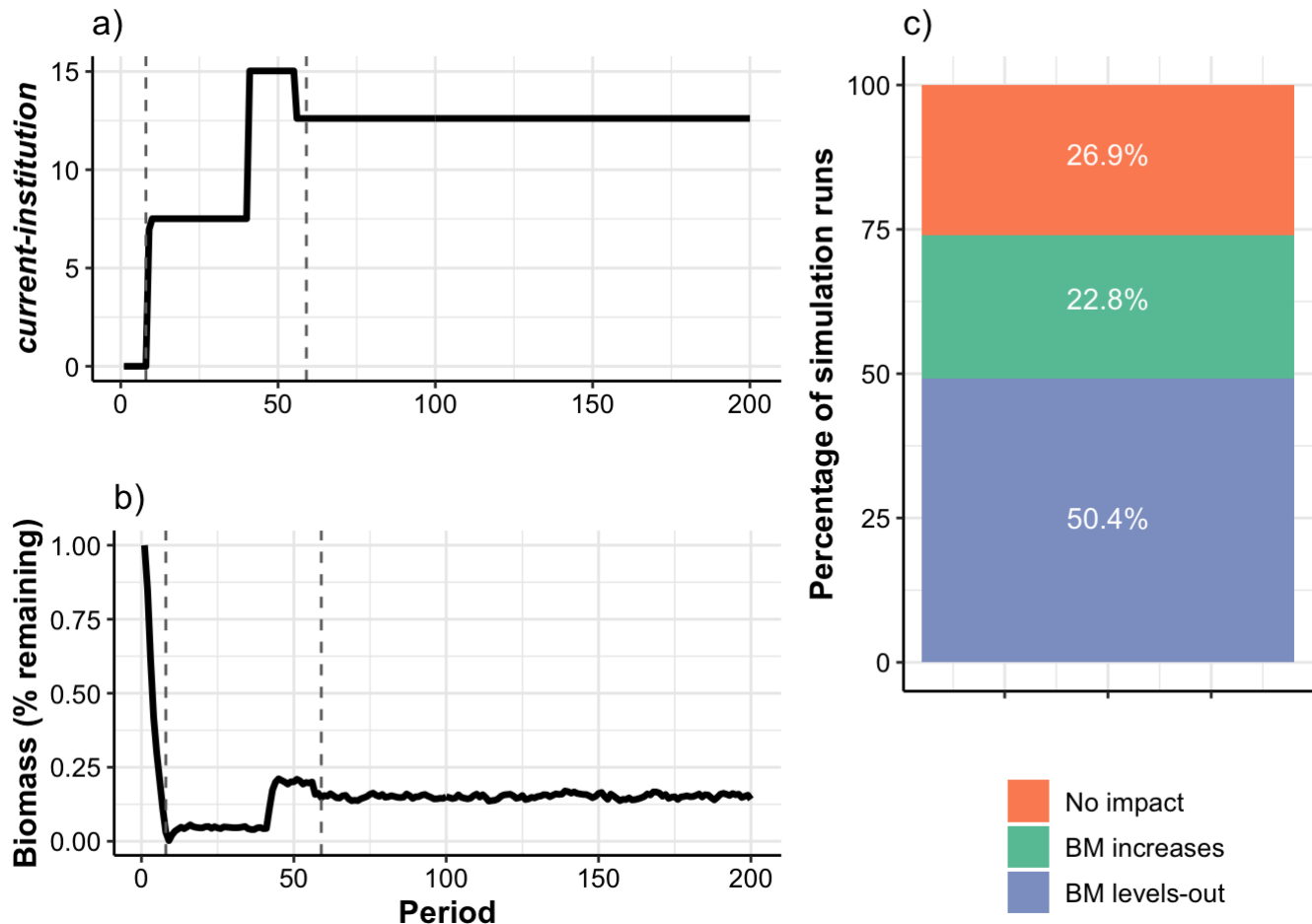
## # A tibble: 3 x 2
##   prop `n()`
##   <dbl> <int>
## 1  24.8     1
## 2  26.1     1
## 3  49.2     1

```

```

viz_jumpsAll=grid.arrange(arrangeGrob(arrangeGrob(arrangeGrob(v2+ggtitle("a")),v3+ggtitl
e("b")), ncol=1, nrow=2)), v4+ggtitle("c"), ncol=2, widths=c(3,2))

```



```
ggsave("viz_jumpsAll.pdf", viz_jumpsAll, width = 5, height = 5, units = "in")
```

mon and sanc boxplots

```
ax_sanc = expression(bolditalic(sanction-level))
ax_mon = expression(bolditalic(monitored-level))

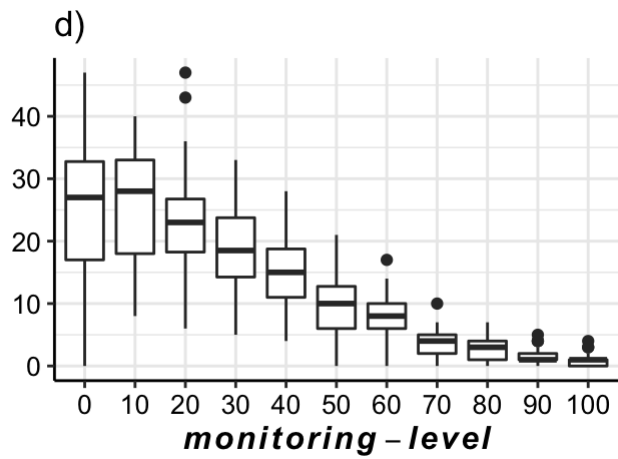
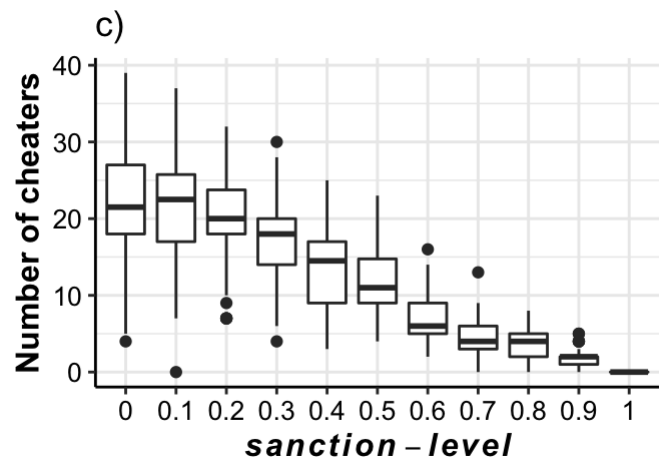
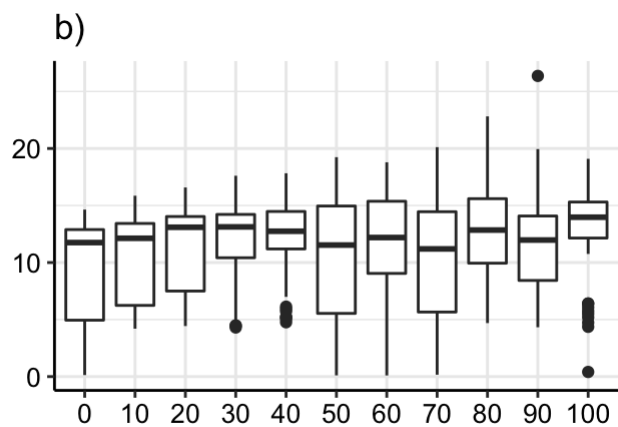
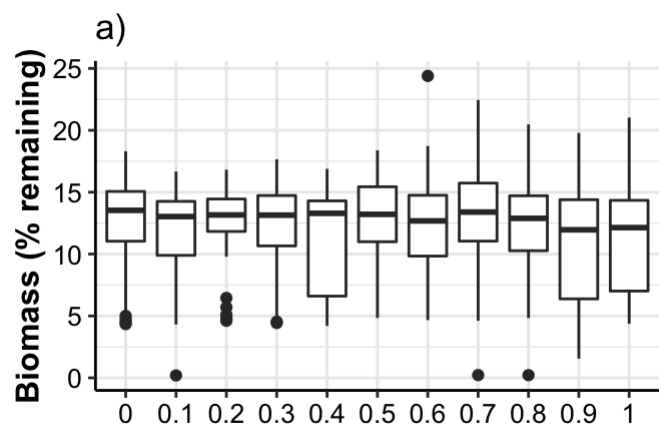
sancBM_graph<-sanctionData%>%ggplot(aes(x=as.factor((as.numeric(variableValue)-1)/10), y=
TotalBM*100))+
  geom_boxplot()+
  labs(x="", y=axlab_bm)+
  setGraphs

# increased monitoring decreases cheaters (yay!)
monBM_graph<-monitoringData%>%ggplot(aes(x=variableValue, y=TotalBM*100))+
  geom_boxplot()+
  labs(x="", y="")+
  setGraphs

# increased sanctioning decreases cheaters (yay!)
sancCheaters_graph<-sanctionData%>%ggplot(aes(x=as.factor((as.numeric(variableValue)-1)/
10), y=numCheaters))+
  geom_boxplot()+
  labs(x=ax_sanc, y="Number of cheaters")+
  setGraphs

# increased monitoring decreases cheaters (yay!)
monCheaters_graph<-monitoringData%>%ggplot(aes(x=variableValue, y=numCheaters))+
  geom_boxplot()+
  labs(x=ax_mon, y="")+
  setGraphs

viz_monSanc=grid.arrange(sancBM_graph+ggtitle("a"), monBM_graph+ggtitle("b"), sancChea
ters_graph+ggtitle("c"), monCheaters_graph+ggtitle("d"), nrow=2)
```



```
ggsave("viz_monSanc_boxplots.pdf", viz_monSanc, width = 5.75, height = 4.4, units = "in"
)
```