

コリスチン発生評価モデル取扱い説明書

2019 年 3 月 29 日（金）

モデル作成者： 酪農学園大学 獣医疫学教授 蒔田浩平

作成補助者： 酪農学園大学大学院生 藤本悠理・三山豪士

本モデルは、日本の出荷直前肉豚に占める *mcr* 保有コリスチン耐性大腸菌の腸管内優勢豚の割合をリスクと定義し、そのリスクを定量的に評価するものです。シミュレーションにより、2017 年、2018 年現在のリスクと対策の効果予測、リスク推移の将来予測をすることが出来ます。確率分布の更新により、ある程度様々な状況の変化に応じたリスクの変化を推定することが出来ます。

本モデルは、無料ソフトである統計ソフト R と Rstudio を用いて作成しました。モデルには、以下 9 枚のワークシート（R ファイル）の作成が必要であり、作成したワークシートを RStudio で開いた後、「モデルシート」の項に記載した指示に従ってコードを読み込むと、モデルを走らせることが出来ます。

モデルコードは、本説明書の以下の表でお示ししたページに記載しています。

問合せ先：酪農学園大学 蒔田浩平 kmakita@rakuno.ac.jp

*ワークシートの種類とモデルコードの記載ページ

ファイル名	説明	記載ページ
Probabilities.R	確率分布	2
Farm_pig.R	農場と豚の作出	10
Sf_mcrp_feedu.R	飼料添加物使用小規模農場	14
Sf_mcrp_feedn.R	飼料添加物不使用小規模農場	22
Mf_mcrp_feedu.R	飼料添加物使用中規模農場	31
Mf_mcrp_feedn.R	飼料添加物不使用中規模農場	39
Lf_mcrp_feedu.R	飼料添加物使用大規模農場	47
Lf_mcrp_feedn.R	飼料添加物不使用大規模農場	55
Modelsheet.R	モデルシート	63

コリスチン リスクモデル コード「確率分布」

ワークシート「Probabilities.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な確率分布を作出するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Probabilities.R」という名前で保存し、「Shift」＋「Ctrl」＋「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一字一文字目に#を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

1. *mcr*保有コリスチン耐性大腸菌侵入農場の割合

1.1. コリスチン含有飼料添加物使用 *mcr*保有コリスチン耐性大腸菌侵入農場におけるプラスミド性コリスチン耐性大腸菌優勢豚の割合

農場実験データ：mean 31.0%, median 30.6% (95%CI: 18.0%-45.6%)。

```
pf2<-numeric(5000)
for (i in 1:5000){
  pf2[i]<-rbeta(1,12.851,28.739)
}
```

1.2. JVARM での *mcr*保有コリスチン耐性大腸菌分離率 (>MIC4mg/L) JVARM データ

Mean 4.5%, median 4.5% (95%CI: 3.1%-6.2%)。# 以降は *mcr* 遺伝子の農場からの消失の効果推定に、「#」を外して用いる。

```
jmp<-numeric(5000)
for (i in 1:5000){
  jmp[i]<-rbeta(1,31+1,706-31+1)*#(0.9^5)
}
```

JVARM データで MIC2mg/L 以下の *mcr* 保有大腸菌も含む場合。

Mean 6.9%, median 6.9% (95%CI: 5.2%-8.9%)。

```
jmp2<-numeric(5000)
for (i in 1:5000){
  jmp2[i]<-rbeta(1,48+1,706-48+1)
}
```

1.3. 真に農場に *mcr* が侵入している割合 PTF

Mean 15.5%, median 14.8% (95%CI: 8.6% - 26.9%)

```
pf1<-numeric(5000)
for (i in 1:5000){
  pf1[i]<-jmp[i]/pf2[i]
}
```

JVARM データで MIC2mg/L 以下の *mcr* 保有大腸菌も含む場合。

```
pf1a<-numeric(5000)
for (i in 1:5000){
  pf1a[i]<-jmp2[i]/pf2[i]
}
```

2. 2017 年度にコリスチン含有飼料添加物を使用していた農場の割合

2017 年度質問票調査では 93%であった。モデルで for ループを確実に回すのに、5000 個一度に生成したいので、若干幅を持たせて無情報分布から抽出。下段は 2018 年度で農場割合は 0%。#を入れ替えて切り替える。

```
pfa<-numeric(5000)
for (i in 1:5000){
  pfa[i]<-runif(1,0.92,0.94) # Year 2017
  # pfa[i]<-runif(1,0.0,0.001) # Year 2018
}
```


3. 浮腫病

3.1.1. 飼料添加物不使用小規模農場で浮腫病が発生する確率

2018 年度質問票調査。#以降は、浮腫病低減による効果推定に用いる（#を外す）。

```
psffned<-numeric(5000)
for (i in 1:5000){
  psffned[i]<-rbeta(1,0.907,12.414)*#0.2
}
```

3.1.2. 飼料添加物使用小規模農場で浮腫病が発生する確率

```
psffued<-numeric(5000)
for (i in 1:5000){
  psffued[i]<-rbeta(1,0.345,70.463)*#0.2
}
```

3.2.1. 飼料添加物不使用中規模農場で浮腫病が発生する確率

```
pmffned<-numeric(5000)
for (i in 1:5000){
  pmffned[i]<-rbeta(1,14.960,254.329)*#0.2
}
```

3.2.2. 飼料添加物使用中規模農場で浮腫病が発生する確率

```
pmffued<-numeric(5000)
for (i in 1:5000){
  pmffued[i]<-rbeta(1,4.295,210.557)*#0.2
}
```

3.3.1. 飼料添加物不使用大規模農場で浮腫病が発生する確率

```
plffned<-numeric(5000)
for (i in 1:5000){
  plffned[i]<-rbeta(1,24.601,242.539)*#0.2
}
```

3.3.2. 飼料添加物使用大規模農場で浮腫病が発生する確率


```
plffued<-numeric(5000)
for (i in 1:5000){
  plffued[i]<-rbeta(1,35.170,305.776)*0.2
}
```

3.4.1. 小規模農場で浮腫病が発生した時の農場内離乳豚発生割合

```
psped<-numeric(5000)
for (i in 1:5000){
  psped[i]<-rbeta(1,1.185,12.660)
}
```

3.4.2. 中規模農場で浮腫病が発生した時の農場内離乳豚発生割合

```
pmped<-numeric(5000)
for (i in 1:5000){
  pmped[i]<-rbeta(1,0.288,1.243)
}
```

3.4.3. 大規模農場で浮腫病が発生した時の農場内離乳豚発生割合

```
plped<-numeric(5000)
for (i in 1:5000){
  plped[i]<-rbeta(1,0.360,1.453)
}
```

4. 細菌性下痢症

4.1.1. 飼料添加物不使用小規模農場で下痢症が発生する確率

下痢症を低減する対策を実施した効果を推定する時には#を外します。

```
psffndi<-numeric(5000)
for (i in 1:5000){
  psffndi[i]<-rbeta(1,0.688,8.288)*0.8
}
```

4.1.2. 飼料添加物使用小規模農場で下痢症が発生する確率

```
psffudi<-numeric(5000)
for (i in 1:5000){
  psffudi[i]<-rbeta(1,1.387,43.623)*0.8
}
```

4.2.1. 飼料添加物不使用中規模農場で下痢症が発生する確率

```
pmffndi<-numeric(5000)
for (i in 1:5000){
  pmffndi[i]<-rbeta(1,41.237,113.062)*0.8
}
```

4.2.2. 飼料添加物使用中規模農場で下痢症が発生する確率

```
pmffudi<-numeric(5000)
for (i in 1:5000){
  pmffudi[i]<-rbeta(1,28.128,128.691)*0.8
}
```

4.3.1. 飼料添加物不使用大規模農場で下痢症が発生する確率

```
plffndi<-numeric(5000)
for (i in 1:5000){
  plffndi[i]<-rbeta(1,86.945,154.822)*0.8
}
```

4.3.2. 飼料添加物使用中規模農場で下痢症が発生する確率

```
plffudi<-numeric(5000)
for (i in 1:5000){
  plffudi[i]<-rbeta(1,111.600,166.215)*0.8
}
```

4.4.1. 下痢症発生時の小規模農場における離乳豚の農場内発生割合

```
pspdi<-numeric(5000)
for (i in 1:5000){
  pspdi[i]<-rbeta(1,1.185,12.660)
}
```

4.4.2. 下痢症発生時の中規模農場における離乳豚の農場内発生割合

```
pmpdi<-numeric(5000)
for (i in 1:5000){
  pmpdi[i]<-rbeta(1,0.866,6.476)
}
```

4.4.3. Proportion of weaning diarrhea pigs in a large scale farm

```
plpdi<-numeric(5000) # Fixed # Questionnaire survey (Makita et al.)
for (i in 1:5000){
  plpdi[i]<-rbeta(1,111.600,166.215)
}
```

5. 治療としてのコリスチンの使用

5.1. 浮腫病発生時のコリスチンによる治療確率

2018 年度の治療確率を用いる時は、#を用いて 2017 年度データと切り替えます。豚房毎の治療をする場合は、2 割の豚にコリスチンを投与するという仮定で#を取って 0.2 を掛けます。

```
pfedcu<-numeric(5000)
for (i in 1:5000){
  pfedcu[i]<-rbeta(1,64.726,53.147)*#0.2 # 2017 data
# pfedcu[i]<-rbeta(1,73.076,91.425)*#0.2 #2018 data
}
```

5.2. 下痢症発生時のコリスチンによる治療確率

5.1 に準ず。

```
pfdicu<-numeric(5000)
for (i in 1:5000){
  pfdicu[i]<-rbeta(1,63.212,105.990)*#0.2 # 2017 data
# pfdicu[i]<-rbeta(1,88.762,193.777)*# 2018 data
}
```


6. *mcr*保有コリスチン耐性大腸菌の優勢割合

6.1. 飼料添加物を使用していない（2018 年度）農場で浮腫病または下痢症発生時にコリスチンを用いた治療を実施した場合の *mcr*保有コリスチン耐性大腸菌の優勢割合

下段の#以降は、豚房毎の治療をする場合。

```
pmpfneddlicudom<-numeric(5000)
for (i in 1:5000){
  pmpfneddlicudom[i]<-rbeta(1,31+1,35-31+1)
# pmpfneddlicudom[i]<-rbeta(1,31+1,35-31+1)*min(0.2+(1-0.2)*rbeta(1,1
4.245,28.937),1) # Pen Level treatment
}
```

6.2. 飼料添加物を使用している（2017 年度）農場で浮腫病または下痢症発生時にコリスチンを用いた治療を実施した場合の *mcr*保有コリスチン耐性大腸菌の優勢割合

下段の#以降は、豚房毎の治療をする場合。

```
pmpfueddicudom<-numeric(5000)
for (i in 1:5000){
  pmpfueddicudom[i]<-rbeta(1,22+1,22-22+1)
# pmpfueddicudom[i]<-rbeta(1,22+1,22-22+1)*min((0.2+(1-0.2)*rbeta(1,1
2.851,28.739)),1) # Pen Level treatment
}
```

6.3. 飼料添加物を使用していない（2018 年度）農場で浮腫病または下痢症発生したがコリスチンによる治療を実施しなかった場合の出荷直前の *mcr*保有コリスチン耐性大腸菌の優勢割合

```
pmpfneddlicndom<-numeric(5000)
for (i in 1:5000){
  pmpfneddlicndom[i]<-rbeta(1,14.245,28.937)
}
```

6.4. 飼料添加物を使用している（2017 年度）農場で浮腫病または下痢症発生したがコリスチンによる治療を実施しなかった場合の出荷直前の *mcr*保有コリスチン耐性大腸菌の優勢割合

```
pmpfueddicndom<-numeric(5000)
for (i in 1:5000){
  pmpfueddicndom[i]<-rbeta(1,12.851,28.739)
}
```

6.5. 飼料添加物を使用していない（2018 年度）農場で浮腫病・下痢症の発生なく、コリスチンによる治療歴のない場合の出荷直前の *mcr* 保有コリスチン耐性大腸菌の優勢割合

```
pmpfnnoeddicndom<-numeric(5000)
for (i in 1:5000){
  pmpfnnoeddicndom[i]<-rbeta(1,14.245,28.937)
}
```

6.6. 飼料添加物を使用している（2017 年度）農場で浮腫病・下痢症の発生なく、コリスチンによる治療歴のない場合の出荷直前の *mcr* 保有コリスチン耐性大腸菌の優勢割合

```
pmpfunoeddicndom<-numeric(5000)
for (i in 1:5000){
  pmpfunoeddicndom[i]<-rbeta(1,12.851,28.739)
}
```

7. 耐性選択後の優勢維持率

コリスチンによる治療の後、*mcr* 保有コリスチン耐性大腸菌が選択され腸管内で優勢となった後に、優勢状態を維持して出荷直前に至る確率

```
pmaint<-0.8
```

コリスチン リスクモデル コード「農場と豚の作出」

ワークシート「Farm_pig.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、1000 戸の仮想農場と、そこで全ての母豚から一回生まれる肉用豚の作出に関するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Pig_farm.R」という名前で保存し、「Shift」+「Ctrl」+「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

各小規模農場の母豚数の設定

農場数、母豚数の上限と下限はワークシート「Modelsheer.R」で設定している。

```
sfarm<-trunc(runif(n_sfarms,n_spig_lower,n_spig_upper))
```

各中規模農場の母豚数の設定

農場数、母豚数の上限と下限はワークシート「Modelsheer.R」で設定している。

```
mfarm<-trunc(runif(n_mfarms,n_mpig_lower,n_mpig_upper))
```

各大規模農場の母豚数の設定

農場数、母豚数の上限と下限はワークシート「Modelsheer.R」で設定している。

```
lfarm<-trunc(runif(n_lfarms,n_lpig_lower,n_lpig_upper))
```


1. 小規模農場の設定

1.1. *mcr* 侵入小規模農場数

農場に *mcr* が侵入している真の割合はワークシート「Probabilities.R」で設定している。

```
nsmp<-round(pf1*length(sfarm))
```

1.2. 小規模農場の ID 番号

```
listsfarm<-c(1:n_sfarms)
```

1.3. *mcr* が侵入した小規模農場の ID 番号

```
listsmp<-function(nsmp){  
  flistsmp<-sample(listsfarm,size=nsmp,replace=FALSE)  
  return(flistsmp)  
}
```

1.4. 各 *mcr* 侵入小規模農場における全母豚からの産仔数

```
pigsmp<-function(nsmp){  
  fpigsmp<-sfarm[listsmp(nsmp)]*litter  
  return(fpigsmp)  
}
```

2. 中規模農場の設定

2.1. *mcr* 侵入中規模農場数

農場に *mcr* が侵入している真の割合はワークシート「Probabilities.R」で設定している。

```
nmmp<-round(pf1*length(mfarm))
```

2.2. 中規模農場の ID 番号

```
listmfarm<-c(1:n_mfarms)
```

2.3. *mcr* が侵入した中規模農場の ID 番号

```
listmmp<-function(nmmp){  
  flistmmp<-sample(listmfarm,size=nmmp,replace=FALSE)  
  return(flistmmp)  
}
```

2.4. 各 *mcr* 侵入中規模農場における全母豚からの産仔数

```
pigmmp<-function(nmmp){  
  fpigmmp<-mfarm[listmmp(nmmp)]*litter  
  return(fpigmmp)  
}
```

3. 大規模農場の設定

3.1. *mcr* 侵入大規模農場数

農場に *mcr* が侵入している真の割合はワークシート「Probabilities.R」で設定している。

```
nlmp<-round(pf1*length(lfarm))
```

3.2. 大規模農場の ID 番号

```
listlfarm<-c(1:n_lfarms)
```

3.3. *mcr* が侵入した大規模農場の ID 番号

```
listlmp<-function(nlmp){  
  flistlmp<-sample(listlfarm,size=nlmp,replace=FALSE)  
  return(flistlmp)  
}
```

3.4. 各 *mcr* 侵入大規模農場における全母豚からの産仔数

```
piglmp<-function(nlmp){  
  fpiglmp<-lfarm[listlmp(nlmp)]*litter  
  return(fpiglmp)  
}
```


コリスチン リスクモデル コード「飼料添加物使用小規模 *mcr* 侵入農場」

ワークシート「Sf_mcrp_feedu.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、飼料添加物使用小規模 *mcr* 侵入農場における耐性選択のダイナミクスを表現するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Sf_mcrp_feedu.R」という名前で保存し、「Shift」＋「Ctrl」＋「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

飼料添加物使用小規模 *mcr* 侵入農場のリスト作成

```
listsmpfu<-function(pfa,lstsmp){  
  flistsmpfu<-sample(lstsmp,size=round(length(lstsmp)*pfa))  
  return(flistsmpfu)  
}
```

飼料添加物使用小規模 *mcr* 侵入農場において母豚から生まれた子豚の総数

（注：モデルには不使用）

```
pigsmpfu<-function(lstsmpfu){  
  fpigsmpfu<-sfarm[lstsmpfu]*litter  
  return(fpigsmpfu)  
}
```

1. 浮腫病が発生し、コリスチンによる治療が行われた農場

1.1. 浮腫病が発生した飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfued<-function(psffued,lstsmpfu){  
  flistsmpfued<-sample(lstsmpfu,size=round(length(lstsmpfu)*psffued))  
  return(flistsmpfued)  
}
```

1.2. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfuedcu<-function(pfedcu,lstsmpfued){  
  flistsmpfuedcu<-sample(lstsmpfued,  
                          size=round(length(lstsmpfued)*pfedcu))  
  return(flistsmpfuedcu)  
}
```

1.3. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場で死んだ豚の頭数（発症豚は全頭死亡）

```
deathsmfuedcu<-function(psped,lstsmpfuedcu){  
  fdeathsmfuedcu<-rbinom(length(lstsmpfuedcu),  
                          sfarm[lstsmpfuedcu]*litter,psped)  
  return(fdeathsmfuedcu)  
}
```

1.4. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfuedcudom<-function(psped,pmpfueddicudom,lstsmpfuedcu){  
  fpigsmpfuedcudom<-rbinom(length(lstsmpfuedcu),  
                             sfarm[lstsmpfuedcu]*litter,  
                             (1-psped)*pmpfueddicudom*pmaint)  
  return(fpigsmpfuedcudom)  
}
```


2. 浮腫病が発生したがコリスチンによる治療をしない農場

2.1. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfuedcn<-function(pfedcu,listsmpfued){  
  flistsmpfuedcn<-sample(listsmfued,  
                           size=round(length(listsmfued)*(1-pfedcu)))  
  return(flistsmpfuedcn)  
}
```

2.2. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場で死亡した豚の頭数

```
deathsmfuedcn<-function(psped,listsmpfuedcn){  
  fdeathsmfuedcn<-rbinom(length(listsmfuedcn),  
                           sfarm[listsmpfuedcn]*litter,psped)  
  return(fdeathsmfuedcn)  
}
```

2.3. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfuedcndom<-function(psped,pmpfueddicndom,listsmpfuedcn){  
  fpigsmpfuedcndom<-rbinom(length(listsmfuedcn),  
                             sfarm[listsmpfuedcn]*litter,  
                             (1-psped)*pmpfueddicndom)  
  return(fpigsmpfuedcndom)  
}
```

3. 細菌性下痢症が発生しコリスチンによる治療が実施された農場

3.1. 下痢症の発生した飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfudi<-function(psffudi,listsmpfu){  
  flistsmpfudi<-sample(listsmfpu,size=round(length(listsmfpu)*psffudi))
```



```
return(flistsmpfudi)
}
```

3.2. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfudicu<-function(pfdicu,listsmpfudi){
  flistsmpfudicu<-sample(listsmfudi,size=round(length(listsmfudi)*pfdi
cu))
  return(flistsmpfudicu)
}
```

3.3. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfudicudom<-function(pfdicu,pmpfueddicudom,listsmpfudi){
  listsmpfudicu<-listsmpfudicu(pfdicu,listsmpfudi)
  fpigsmpfudicudom<-rbinom(length(listsmfudicu),
                             sfarm[listsmpfudicu]*litter,
                             pmpfueddicudom*pmaint)
  return(fpigsmpfudicudom)
}
```

4. 細菌性下痢症が発生したがコリスチンによる治療が実施されなかった農場

4.1. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfudicn<-function(pfdicu,listsmpfudi){
  flistsmpfudicn<-sample(listsmfudi,
                          size=round(length(listsmfudi)*(1-pfdicu)))
  return(flistsmpfudicn)
}
```

4.2. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfudicndom<-function(pfdicu,pmpfueddicndom,lstsmpfudi){  
  lstsmpfudicn<-listsmpfudicn(pfdicu,lstsmpfudi)  
  fpigsmpfudicndom<-rbinom(length(lstsmpfudicn),  
                             sfarm[lstsmpfudicn]*litter,  
                             pmpfueddicndom)  
  return(fpigsmpfudicndom)  
}
```

5. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた農場

5.1. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfueddicu<-function(psffudi,lstsmpfuedcu){  
  flistsmpfueddicu<-sample(lstsmpfuedcu,  
                            size=round(length(lstsmpfuedcu)*psffudi))  
  return(flistsmpfueddicu)  
}
```

5.2. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物使用小規模 *mcr* 侵入農場で、上記 1 と 3 でダブルカウントしている出荷時 *mcr* 保有コリスチン耐性大腸菌優勢豚の頭数

```
pigsmpfucuooverlap<-function(psffudi,pmpfueddicu,lstsmpfuedcu){  
  lstsmpfueddicu<-listsmpfueddicu(psffudi,lstsmpfuedcu)  
  fpigsmpfucuooverlap<-rbinom(length(lstsmpfueddicu),  
                               sfarm[lstsmpfueddicu]*litter,  
                               pmpfueddicu*pmain)  
  return(fpigsmpfucuooverlap)  
}
```


6. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が実施され

なかった農場

6.1. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfueddicn<-function(psffudi,lstsmpfuedcn){  
  flistsmpfueddicn<-sample(lstsmpfuedcn,  
                             size=round(length(lstsmpfuedcn)*psffudi))  
  return(flistsmpfueddicn)  
}
```

6.2. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfucnoverlap<-function(psffudi,pmpfueddicndom,lstsmpfuedcn){  
  lstsmpfueddicn<-listsmpfueddicn(psffudi,lstsmpfuedcn)  
  fpigsmpfucnoverlap<-rbinom(length(lstsmpfueddicn),  
                               sfarm[lstsmpfueddicn]*litter,  
                               pmpfueddicndom)  
  return(fpigsmpfucnoverlap)  
}
```

7. 浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった農場

浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった飼料添加物使用小規模 *mcr* 侵入農場で、出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfunoeddicndom<-function(psffued,psffudi,pmpfunoeddicndom,lstsmpfu){  
  fpigsmpfunoeddicndom<-round(sum(sfarm[lstsmpfu]*litter)*  
                               (1-(psffued+psffudi-psffued*psffudi))*pmp  
funoeddicndom)
```



```
# fpigsmpfunoeddicndom<-rbinom(round(length(listsmpfu(nsmf,pfa))*ma  
x(0.001,(1-(psffued+psffudi-psffued*psffudi)))),  
# sfarm[listsmpfu(nsmf,pfa)]*litter,  
# pmpfunoeddicndom)  
return(fpigsmpfunoeddicndom)  
}
```

8. 合計頭数

8.1. 飼料添加物使用小規模 *mcr* 侵入農場で生産され出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の合計頭数

```
totalmcrpigsmpfu<-function(psffued,psffudi,pfdicu,psped,  
pmpfueddicndom,pmpfueddicndom,pmpfunoeddicndom,  
listsmpfu,listsmpfuedcu,listsmpfuedcn){  
listsmpfudi<-listsmpfudi(psffudi,listsmpfu)  
ftotalmcrpigsmpfu<-sum(pigsmpfuedcndom(psped,pmpfueddicndom,listsmpfu  
edcu))+  
sum(pigsmpfuedcndom(psped,pmpfueddicndom,listsmpfuedcn))+  
sum(pigsmpfudicndom(pfdicu,pmpfueddicndom,listsmpfudi))+  
sum(pigsmpfudicndom(pfdicu,pmpfueddicndom,listsmpfudi))-  
sum(pigsmpfucuooverlap(psffudi,pmpfueddicndom,listsmpfuedcu))-  
sum(pigsmpfucnooverlap(psffudi,pmpfueddicndom,listsmpfuedcn))+  
pmpfunoeddicndom(psffued,psffudi,pmpfunoeddicndom,listsmpfu)  
return(ftotalmcrpigsmpfu)  
}
```

8.2. 飼料添加物使用小規模 *mcr* 侵入農場における浮腫病による合計死亡頭数

```
totaldeathsmpfu<-function(psped,listsmpfuedcu,listsmpfuedcn){  
ftotaldeathsmpfu<-sum(deathsmpfuedcu(psped,listsmpfuedcu))+  
sum(deathsmpfuedcn(psped,listsmpfuedcn))  
return(ftotaldeathsmpfu)  
}
```

8.3. 飼料添加物使用小規模 *mcr* 侵入農場で生産された健康豚の合計頭数（*pmcrpigsmpfu* と *integratedslpigmp* に使用されているが、リスク推定には直接使用されていない）

```
totalslsmfu<-function(pfedcu,psped,lstsmfu,lstsmfued){  
  lstsmfuedcu<-listsmfuedcu(pfedcu,lstsmfued)  
  lstsmfuedcn<-setdiff(lstsmfued,lstsmfuedcu)  
  ftotalslsmfu<-sum(sfarm[lstsmfu]*litter)-  
  totaldeathsmfu(psped,lstsmfuedcu,lstsmfuedcn)  
  return(ftotalslsmfu)  
}
```

9. このグループのリスク

出荷時の *mcr* 保有コリスチン耐性大腸菌優勢豚割合（詳細なモニタリング用。モデルには未使用。）

```
pmcrpigsmpfu<-function(nsmf,pfa,psffued,psffudi,pfedcu,pfdicu,psped,  
  pmpfneddudom,pmpfueddicndom,pmpfunoeddudom){  
  lstsmf<-listsmf(nsmf)  
  lstsmfu<-listsmfu(pfa,lstsmf)  
  lstsmfued<-listsmfued(psffued,lstsmfu)  
  lstsmfuedcu<-listsmfuedcu(pfedcu,lstsmfued)  
  lstsmfuedcn<-setdiff(lstsmfued,lstsmfuedcu)  
  fpmcrpigsmpfu<-totalmcrpigsmpfu(psffued,psffudi,pfdicu,psped,  
    pmpfneddudom,pmpfueddicndom,pmpfunoeddudom,  
    dicndom,lstsmfu,lstsmfuedcu,lstsmfuedcn)/  
  totalslsmfu(pfedcu,psped,lstsmfu,lstsmfued)  
  return(fpmcrpigsmpfu)  
}
```


コリスチン リスクモデル コード「飼料添加物不使用小規模 *mcr* 侵入農場」

ワークシート「Sf_mcrp_feedn.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、飼料添加物不使用小規模 *mcr* 侵入農場における耐性選択のダイナミクスを表現するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Sf_mcrp_feedn.R」という名前で保存し、「Shift」+「Ctrl」+「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

飼料添加物不使用小規模 *mcr* 侵入農場のリスト作成

```
listsmpfn<-function(pfa,lstsmp){  
  flistsmpfn<-sample(lstsmp,size=max(1,round(length(lstsmp)*(1-pfa))))  
  # flistsmpfn<-sample(lstsmp,size=round(length(lstsmp)*(1-pfa)))  
  return(flistsmpfn)  
}
```

飼料添加物不使用小規模 *mcr* 侵入農場において母豚から生まれた子豚の総数

(注：モデルには不使用)

```
pigsmpfn<-function(lstsmpfn){  
  fpigsmpfn<-sfarm[lstsmpfn]*litter  
  return(fpigsmpfn)  
}
```


1. 浮腫病が発生し、コリスチンによる治療が行われた農場

1.1. 浮腫病の発生した飼料添加物不使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfned<-function(psffned,lstsmpfn){  
  flistsmpfned<-sample(lstsmpfn,size=round(length(lstsmpfn)*psffned))  
  return(flistsmpfned)  
}
```

1.2. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfnedcu<-function(pfedcu,lstsmpfned){  
  flistsmpfnedcu<-sample(lstsmpfned,  
                          size=round(length(lstsmpfned)*pfedcu))  
  return(flistsmpfnedcu)  
}
```

1.3. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場で死んだ豚の頭数（発症豚は全頭死亡）

```
deathsmpfnedcu<-function(psped,lstsmpfnedcu){  
  fdeathsmpfnedcu<-rbinom(length(lstsmpfnedcu),  
                           sfarm[lstsmpfnedcu]*litter,psped)  
  return(fdeathsmpfnedcu)  
}
```

1.4. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfnedcudom<-function(psped,pmpfneddicudom,lstsmpfnedcu){  
  fpigsmpfnedcudom<-rbinom(length(lstsmpfnedcu),  
                             sfarm[lstsmpfnedcu]*litter,  
                             (1-psped)*pmpfneddicudom*pmaint)  
  return(fpigsmpfnedcudom)  
}
```

2. 浮腫病が発生したがコリスチンによる治療をしない農場

2.1. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用小規模
mcr 侵入農場のリスト (ID 番号)

```
listsmpfnedcn<-function(pfedcu,listsmpfned){  
  flistsmpfnedcn<-sample(listsmpfned,  
                           size=round(length(listsmpfned)*(1-pfedcu)))  
  return(flistsmpfnedcn)  
}
```

2.2. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用小規模
mcr 侵入農場で死亡した豚の頭数

```
deathsmpfnedcn<-function(psped,listsmpfnedcn){  
  fdeathsmpfnedcn<-rbinom(length(listsmpfnedcn),  
                            sfarm[listsmpfnedcn]*litter,psped)  
  return(fdeathsmpfnedcn)  
}
```

2.3. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不
使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢で
ある豚の頭数

```
pigsmpfnedcndom<-function(psped,pmpfneddicndom,listsmpfnedcn){  
  fpigsmpfnedcndom<-rbinom(length(listsmpfnedcn),  
                             sfarm[listsmpfnedcn]*litter,  
                             (1-psped)*pmpfneddicndom)  
  return(fpigsmpfnedcndom)  
}
```


3. 細菌性下痢症が発生しコリスチンによる治療が実施された農場

3.1. 下痢症の発生した飼料添加物不使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfndi<-function(psffndi,lstsmpfn){  
  flistsmpfndi<-sample(lstsmpfn,size=round(length(lstsmpfn)*psffndi))  
  return(flistsmpfndi)  
}
```

3.2. 下痢症が発生しコリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfndicu<-function(pfdicu,lstsmpfndi){  
  flistsmpfndicu<-sample(lstsmpfndi,size=round(length(lstsmpfndi)*pfdicu))  
  return(flistsmpfndicu)  
}
```

3.3. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmpfndicudom<-function(pfdicu,pmpfneddlicudom,lstsmpfndi){  
  listsmpfndicu<-listsmpfndicu(pfdicu,lstsmpfndi)  
  fpigsmpfndicudom<-rbinom(length(lstsmpfndicu),  
                             sfarm[lstsmpfndicu]*litter,  
                             pmpfneddlicudom*pmaint)  
  return(fpigsmpfndicudom)  
}
```

4. 細菌性下痢症が発生したがコリスチンによる治療が実施されなかった農場

4.1. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物不使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfndicn<-function(pfdicu,lstsmpfndi){  
  flistsmpfndicn<-sample(lstsmpfndi,
```



```
size=round(length(lstsmfndi)*(1-pfdicu)))  
return(flistsmfndicn)  
}
```

4.2. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物不使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmfndicndom<-function(pfdicu,pmpfneddicndom,lstsmfndi){  
  lstsmfndicn<-listsmfndicn(pfdicu,lstsmfndi)  
  fpigsmfndicndom<-rbinom(length(lstsmfndicn),  
                             sfarm[lstsmfndicn]*litter,  
                             pmpfneddicndom)  
  return(fpigsmfndicndom)  
}
```

5. 浮腫病と細菌性下痢症の両方が発生しコリスチンによる治療が行われた農場

5.1. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場のリスト（ID 番号）

```
listsmfneddicu<-function(psffndi,lstsmfnedcu){  
  flistsmfneddicu<-sample(lstsmfnedcu,  
                           size=round(length(lstsmfnedcu)*psffndi))  
  return(flistsmfneddicu)  
}
```

5.2. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物不使用小規模 *mcr* 侵入農場で、上記 1 と 3 でダブルカウントしている出荷時 *mcr* 保有コリスチン耐性大腸菌優勢豚の頭数

```
pigsmfncuoverlap<-function(psffndi,pmpfneddicudom,lstsmfnedcu){  
  lstsmfneddicu<-listsmfneddicu(psffndi,lstsmfnedcu)  
  fpigsmfncuoverlap<-rbinom(length(lstsmfneddicu),  
                              sfarm[lstsmfneddicu]*litter,  
                              pmpfneddicudom*pmain)
```

```
    return(fpigsmfncuoverlap)
}
```

6. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が実施され

なかった農場

6.1. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物不使用小規模 *mcr* 侵入農場のリスト (ID 番号)

```
listsmpfneddin<-function(psffndi,listsmpfnedcn){
  flistsmpfneddin<-sample(listsmfndcn,
                           size=round(length(listsmfndcn)*psffndi))
  return(flistsmpfneddin)
}
```

6.2. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物不使用小規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigsmfncuoverlap<-function(psffndi,pmpfneddin,listsmpfnedcn){
  listsmpfneddin<-listsmpfneddin(psffndi,listsmpfnedcn)
  fpigsmfncuoverlap<-rbinom(length(listsmfndcn),
                             sfarm[listsmpfneddin]*litter,
                             pmpfneddin)
  return(fpigsmfncuoverlap)
}
```

7. 浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった農場

浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった飼料添加物不使用小規模 *mcr* 侵入農場で、出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数


```
pigsmfnnnoeddicndom<-function(psffned,psffndi,pmpfnnoeddicndom,lstsmf
n){
  fpigsmfnnnoeddicndom<-round(sum(sfarm[lstsmfnn]*litter)*
                                (1-(psffned+psffndi-psffned*psffndi))*pmp
fnnnoeddicndom)
#   fpigsmfnnnoeddicndom<-rbinom(round(length(listsmfnn(nsmf,pfa))*ma
x(0.001,(1-(psffned+psffndi-psffned*psffndi)))),
#                                     sfarm[listsmfnn(nsmf,pfa)]*litter,
#                                     pmpfnnoeddicndom)
  return(fpigsmfnnnoeddicndom)
}
```

8. 合計頭数

8.1. 飼料添加物不使用小規模 *mcr* 侵入農場で生産され出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の合計頭数

```
totalmcrpigsmfnn<-function(psffned,psffndi,pfdicu,psped,
                             pmpfneddicndom,pmpfneddicndom,pmpfnnoeddicndom,
                             lstsmfnn,lstsmfnnedcu,lstsmfnnedcn){
  lstsmfnnndi<-listsmfnnndi(psffndi,lstsmfnn)
  ftotalmcrpigsmfnn<-sum(pigsmfnneddicndom(psped,pmpfneddicndom,lstsmfnn
edcu))+
  sum(pigsmfnneddicndom(psped,pmpfneddicndom,lstsmfnnedcn))+
  sum(pigsmfnnndicndom(pfdicu,pmpfneddicndom,lstsmfnnndi))+
  sum(pigsmfnnndicndom(pfdicu,pmpfneddicndom,lstsmfnnndi))-
  sum(pigsmfnnncuoverlap(psffndi,pmpfneddicndom,lstsmfnnedcu))-
  sum(pigsmfnnncnooverlap(psffndi,pmpfneddicndom,lstsmfnnedcn))+
  pigsmfnnnoeddicndom(psffned,psffndi,pmpfnnoeddicndom,lstsmfnn)
  return(ftotalmcrpigsmfnn)
}
```

8.2. 飼料添加物不使用小規模 *mcr* 侵入農場における浮腫病による合計死亡頭数


```
totaldeathsmfn<-function(psped,lstsmfnedcu,lstsmfnedcn){  
  ftotaldeathsmfn<-sum(deathsmfnedcu(psped,lstsmfnedcu))+  
  sum(deathsmfnedcn(psped,lstsmfnedcn))  
  return(ftotaldeathsmfn)  
}
```

8.3. 飼料添加物不使用小規模 *mcr* 侵入農場で生産された健康豚の合計頭数

(*pmcrpigsmpfn* と *integratedslpigmp* に使用されているが、リスク推定には直接使用されていない)

```
totalslsmfn<-function(pfedcu,psped,lstsmfn,lstsmfned){  
  lstsmfnedcu<-listsmfnedcu(pfedcu,lstsmfned)  
  # lstsmfnedcn<-listsmfnedcn(pfedcu,lstsmfned)  
  lstsmfnedcn<-setdiff(lstsmfned,lstsmfnedcu)  
  ftotalslsmfn<-sum(sfarm[lstsmfn]*litter)-  
  totaldeathsmfn(psped,lstsmfnedcu,lstsmfnedcn)  
  return(ftotalslsmfn)  
}
```

9. このグループのリスク

出荷時の *mcr* 保有コリスチン耐性大腸菌優勢豚割合（詳細なモニタリング用。モデルには未使用。）

```
pmcrpigsmpfn<-function(nsmp,pfa,psffned,psffndi,pfedcu,pfdicu,psped,  
  pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom){  
  lstsmfn<-listsmfn(nsmp)  
  lstsmfnedcu<-listsmfnedcu(pfa,lstsmfn)  
  lstsmfnedcn<-listsmfnedcn(psffned,lstsmfnedcu)  
  lstsmfnedcu<-listsmfnedcu(pfedcu,lstsmfned)  
  # lstsmfnedcn<-listsmfnedcn(pfedcu,lstsmfned)  
  lstsmfnedcn<-setdiff(lstsmfned,lstsmfnedcu)  
  fpmcrpigsmpfn<-totalmcrpigsmpfn(psffned,psffndi,pfdicu,psped,  
    pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,lstsmfn,  
    lstsmfnedcu,lstsmfnedcn)/
```

```
totalslmpfn(pfedcu,psped,1stsmprfn,1stsmprfnd)  
return(fpmcrpigsmpfn)  
}
```

コリスチン リスクモデル コード「飼料添加物使用中規模 *mcr* 侵入農場」

ワークシート「Mf_mcrp_feedu.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、飼料添加物使用中規模 *mcr* 侵入農場における耐性選択のダイナミクスを表現するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Mf_mcrp_feedu.R」という名前で保存し、「Shift」+「Ctrl」+「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一字一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

飼料添加物使用中規模 *mcr* 侵入農場のリスト作成

```
listmmpfu<-function(pfa,lstmmp){  
  flistmmpfu<-sample(lstmmp,size=round(length(lstmmp)*pfa))  
  return(flistmmpfu)  
}
```

飼料添加物使用中規模 *mcr* 侵入農場において母豚から生まれた子豚の総数

（注：モデルには不使用）

```
pigmmpfu<-function(lstmmpfu){  
  fpigmmpfu<-mfarm[lstmmpfu]*litter  
  return(fpigmmpfu)  
}
```


1. 浮腫病が発生し、コリスチンによる治療が行われた農場

1.1. 浮腫病が発生した飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfued<-function(pmffued,lstmmpfu){  
  flistmmpfued<-sample(lstmmpfu,size=round(length(lstmmpfu)*pmffued))  
  return(flistmmpfued)  
}
```

1.2. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfuedcu<-function(pfedcu,lstmmpfued){  
  flistmmpfuedcu<-sample(lstmmpfued,  
                           size=round(length(lstmmpfued)*pfedcu))  
  return(flistmmpfuedcu)  
}
```

1.3. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場で死んだ豚の頭数（発症豚は全頭死亡）

```
deathmmpfuedcu<-function(pmped,lstmmpfuedcu){  
  fdeathmmpfuedcu<-rbinom(length(lstmmpfuedcu),  
                             mfarm[lstmmpfuedcu]*litter,pmped)  
  return(fdeathmmpfuedcu)  
}
```

1.4. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfuedcudom<-function(pmped,pmpfueddicudom,lstmmpfuedcu){  
  fpigmmpfuedcudom<-rbinom(length(lstmmpfuedcu),  
                              mfarm[lstmmpfuedcu]*litter,  
                              (1-pmped)*pmpfueddicudom*pmain)  
  return(fpigmmpfuedcudom)  
}
```

2. 浮腫病が発生したがコリスチンによる治療をしない農場

2.1. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfuedcn<-function(pfedcu,lstmmpfued){  
  flistmmpfuedcn<-sample(lstmmpfued,  
                           size=round(length(lstmmpfued)*(1-pfedcu)))  
  return(flistmmpfuedcn)  
}
```

2.2. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場で死亡した豚の頭数

```
deathmmpfuedcn<-function(pmped,lstmmpfuedcn){  
  fdeathmmpfuedcn<-rbinom(length(lstmmpfuedcn),  
                             mfarm[lstmmpfuedcn]*litter,pmped)  
  return(fdeathmmpfuedcn)  
}
```

2.3. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfuedcndom<-function(pmped,pmpfueddicndom,lstmmpfuedcn){  
  fpigmmpfuedcndom<-rbinom(length(lstmmpfuedcn),  
                              mfarm[lstmmpfuedcn]*litter,  
                              (1-pmped)*pmpfueddicndom)  
  return(fpigmmpfuedcndom)  
}
```

3. 細菌性下痢症が発生しコリスチンによる治療が実施された農場

3.1. 下痢症の発生した飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfudi<-function(pmffudi,lstmmpfu){  
  flistmmpfudi<-sample(lstmmpfu,size=round(length(lstmmpfu)*pmffudi))
```



```
    return(flistmmpfudi)
}
```

3.2. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfudicu<-function(pfdicu,listmmpfudi){
  flistmmpfudicu<-sample(listmmpfudi,size=round(length(listmmpfudi)*pfdi
cu))
  return(flistmmpfudicu)
}
```

3.3. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfudicudom<-function(pfdicu,pmpfueddicudom,listmmpfudi){
  listmmpfudicu<-listmmpfudicu(pfdicu,listmmpfudi)
  fpigmmpfudicudom<-rbinom(length(listmmpfudicu),
                             mfarm[listmmpfudicu]*litter,
                             pmpfueddicudom*pmaint)
  return(fpigmmpfudicudom)
}
```

4. 細菌性下痢症が発生したがコリスチンによる治療が実施されなかった農場

4.1. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfudicn<-function(pfdicu,listmmpfudi){
  flistmmpfudicn<-sample(listmmpfudi,
                          size=round(length(listmmpfudi)*(1-pfdicu)))
  return(flistmmpfudicn)
}
```

4.2. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数


```
pigmmpfudicndom<-function(pfdicu,mpmfueddicndom,lstmmpfudi){  
  lstmmpfudicn<-listmmpfudicn(pfdicu,lstmmpfudi)  
  fpigmmpfudicndom<-rbinom(length(lstmmpfudicn),  
                             mfarm[lstmmpfudicn]*litter,  
                             mpmfueddicndom)  
  return(fpigmmpfudicndom)  
}
```

5. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた農場

5.1. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfueddicu<-function(pmfufudi,lstmmpfuedcu){  
  flistmmpfueddicu<-sample(lstmmpfuedcu,  
                             size=round(length(lstmmpfuedcu)*pmfufudi))  
  return(flistmmpfueddicu)  
}
```

5.2. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物使用中規模 *mcr* 侵入農場で、上記 1 と 3 でダブルカウントしている出荷時 *mcr* 保有コリスチン耐性大腸菌優勢豚の頭数

```
pigmmpfucuooverlap<-function(pmfufudi,mpmfueddicudom,lstmmpfuedcu){  
  lstmmpfueddicu<-listmmpfueddicu(pmfufudi,lstmmpfuedcu)  
  fpigmmpfucuooverlap<-rbinom(length(lstmmpfueddicu),  
                                mfarm[lstmmpfueddicu]*litter,  
                                mpmfueddicudom*pmaint)  
  return(fpigmmpfucuooverlap)  
}
```

6. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が実施され

なかった農場

6.1. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfueddicn<-function(pmffudi,lstmmpfuedcn){  
  flistmmpfueddicn<-sample(lstmmpfuedcn,  
                             size=round(length(lstmmpfuedcn)*pmffudi))  
  return(flistmmpfueddicn)  
}
```

6.2. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfucnoverlap<-function(pmffudi,pmpfueddicndom,lstmmpfuedcn){  
  lstmmpfueddicn<-listmmpfueddicn(pmffudi,lstmmpfuedcn)  
  fpigmmpfucnoverlap<-rbinom(length(lstmmpfueddicn),  
                               mfarm[lstmmpfueddicn]*litter,  
                               pmpfueddicndom)  
  return(fpigmmpfucnoverlap)  
}
```

7. 浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった農場

浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった飼料添加物使用中規模 *mcr* 侵入農場で、出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfunoeddicndom<-function(pmffued,pmffudi,pmpfunoeddicndom,lstmmpfu){  
  fpigmmpfunoeddicndom<-round(sum(mfarm[lstmmpfu]*litter)*  
                               (1-(pmffued+pmffudi-pmffued*pmffudi))*pmp  
funoeddicndom)
```



```
# fpigmpfunoeddicndom<-rbinom(round(length(Listmmpfu(nmmp,pfa))*max  
(0.001,(1-(pmffued+pmffudi-pmffued*pmffudi))))),  
#                                     mfarm[Listmmpfu(nmmp,pfa)]*litter,  
#                                     pmpfunoeddicndom)  
return(fpigmpfunoeddicndom)  
}
```

8. 合計頭数

8.1. 飼料添加物使用中規模 *mcr* 侵入農場で生産され出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の合計頭数

```
totalmcrpigmpfu<-function(pmffued,pmffudi,pfdicu,pmped,  
                           pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
                           lstmmpfu,lstmmpfuedcu,lstmmpfuedcn){  
  lstmmpfudi<-listmmpfudi(pmffudi,lstmmpfu)  
  ftotalmcrpigmpfu<-sum(pigmmpfuedcudom(pmped,pmpfueddicudom,lstmmpfu  
edcu))+  
  sum(pigmmpfuedcndom(pmped,pmpfueddicndom,lstmmpfuedcn))+  
  sum(pigmmpfudicudom(pfdicu,pmpfueddicudom,lstmmpfudi))+  
  sum(pigmmpfudicndom(pfdicu,pmpfueddicndom,lstmmpfudi))-  
  sum(pigmmpfucuooverlap(pmffudi,pmpfueddicudom,lstmmpfuedcu))-  
  sum(pigmmpfucnooverlap(pmffudi,pmpfueddicndom,lstmmpfuedcn))+  
  pigmpfunoeddicndom(pmffued,pmffudi,pmpfunoeddicndom,lstmmpfu)  
  return(ftotalmcrpigmpfu)  
}
```

8.2. 飼料添加物使用中規模 *mcr* 侵入農場における浮腫病による合計死亡頭数

```
totaldeathmmpfu<-function(pmped,lstmmpfuedcu,lstmmpfuedcn){  
  ftotaldeathmmpfu<-sum(deathmmpfuedcu(pmped,lstmmpfuedcu))+  
  sum(deathmmpfuedcn(pmped,lstmmpfuedcn))  
  return(ftotaldeathmmpfu)  
}
```


8.3. 飼料添加物使用中規模 *mcr* 侵入農場で生産された健康豚の合計頭数（*pmcrpigmpfu* と *integratedslpigmp* に使用されているが、リスク推定には直接使用されていない）

```
totalslmpfu<-function(pfedcu,mped,lstmpfu,lstmpfued){  
  lstmpfuedcu<-listmpfuedcu(pfedcu,lstmpfued)  
  # lstmpfuedcn<-listmpfuedcn(pfedcu,lstmpfued)  
  lstmpfuedcn<-setdiff(lstmpfued,lstmpfuedcu)  
  ftotalslmpfu<-sum(mfarm[lstmpfu]*litter)-  
  totaldeathmpfu(mped,lstmpfuedcu,lstmpfuedcn)  
  return(ftotalslmpfu)  
}
```

9. このグループのリスク

出荷時の *mcr* 保有コリスチン耐性大腸菌優勢豚割合（詳細なモニタリング用。モデルには未使用。）

```
pmcrpigmpfu<-function(nmmp,pmffued,pmffudi,pfedcu,pfdicu,mped,  
  pmpfneddlicudom,pmpfueddicndom,pmpfunoeddlicndom){  
  lstmp<-listmp(nmmp)  
  lstmpfu<-listmpfu(pfa,lstmp)  
  lstmpfued<-listmpfued(pmffued,lstmpfu)  
  lstmpfuedcu<-listmpfuedcu(pfedcu,lstmpfued)  
  # lstmpfuedcn<-listmpfuedcn(pfedcu,lstmpfued)  
  lstmpfuedcn<-setdiff(lstmpfued,lstmpfuedcu)  
  fpmcrpigmpfu<-totalmcrpigmpfu(pmffued,pmffudi,pfdicu,mped,  
    pmpfneddlicudom,pmpfunoeddlicndom,lstmpfu,  
    lstmpfuedcu)/  
  totalslmpfu(pmffued,mped,lstmpfu,lstmpfued)  
  return(fpmcrpigmpfu)  
}
```

コリスチン リスクモデル コード「飼料添加物不使用中規模 *mcr* 侵入農場」

ワークシート「Mf_mcrp_feedn.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、飼料添加物不使用中規模 *mcr* 侵入農場における耐性選択のダイナミクスを表現するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Mf_mcrp_feedn.R」という名前で保存し、「Shift」+「Ctrl」+「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一字一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

飼料添加物不使用中規模 *mcr* 侵入農場のリスト作成

```
listmmpfn<-function(pfa,lstmmp){  
  flistmmpfn<-sample(lstmmp,size=max(1,round(length(lstmmp)*(1-pfa))))  
  # flistmmpfn<-sample(lstmmp,size=round(length(lstmmp)*(1-pfa)))  
  return(flistmmpfn)  
}
```

飼料添加物不使用中規模 *mcr* 侵入農場において母豚から生まれた子豚の総数

(注：モデルには不使用)

```
pigmmpfn<-function(lstmmpfn){  
  fpigmmpfn<-mfarm[lstmmpfn]*litter  
  return(fpigmmpfn)  
}
```


1. 浮腫病が発生し、コリスチンによる治療が行われた農場

1.1. 浮腫病の発生した飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfned<-function(pmffned,listmmpfn){  
  flistmmpfned<-sample(listmmpfn,size=round(length(listmmpfn)*pmffned))  
  return(flistmmpfned)  
}
```

1.2. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfnedcu<-function(pfedcu,listmmpfned){  
  flistmmpfnedcu<-sample(listmmpfned,  
                           size=round(length(listmmpfned)*pfedcu))  
  return(flistmmpfnedcu)  
}
```

1.3. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用中規模 *mcr* 侵入農場で死んだ豚の頭数（発症豚は全頭死亡）

```
deathmmpfnedcu<-function(pmped,listmmpfnedcu){  
  fdeathmmpfnedcu<-rbinom(length(listmmpfnedcu),  
                            mfarm[listmmpfnedcu]*litter,pmped)  
  return(fdeathmmpfnedcu)  
}
```

1.4. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfnedcudom<-function(pmped,pmpfneddicudom,listmmpfnedcu){  
  fpigmmpfnedcudom<-rbinom(length(listmmpfnedcu),  
                             mfarm[listmmpfnedcu]*litter,  
                             (1-pmped)*pmpfneddicudom*pmaint)  
  return(fpigmmpfnedcudom)  
}
```

2. 浮腫病が発生したがコリスチンによる治療をしない農場

2.1. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfncdn<-function(pfedcu,lstmmpfncdn){  
  flistmmpfncdn<-sample(lstmmpfncdn,  
                          size=round(length(lstmmpfncdn)*(1-pfedcu)))  
  return(flistmmpfncdn)  
}
```

2.2. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場で死亡した豚の頭数

```
deathmmpfncdn<-function(pmped,lstmmpfncdn){  
  fdeathmmpfncdn<-rbinom(length(lstmmpfncdn),  
                           mfarm[lstmmpfncdn]*litter,pmped)  
  return(fdeathmmpfncdn)  
}
```

2.3. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfncndom<-function(pmped,pmpfncndicndom,lstmmpfncdn){  
  fpigmmpfncndom<-rbinom(length(lstmmpfncdn),  
                           mfarm[lstmmpfncdn]*litter,  
                           (1-pmped)*pmpfncndicndom)  
  return(fpigmmpfncndom)  
}
```

3. 細菌性下痢症が発生しコリスチンによる治療が実施された農場

3.1. 下痢症の発生した飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfndi<-function(pmffndi,lstmmpfn){  
  flistmmpfndi<-sample(lstmmpfn,size=round(length(lstmmpfn)*pmffndi))
```



```
    return(flistmmpfndi)
}
```

3.2. 下痢症が発生しコリスチンによる治療が行われた飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfndicu<-function(pfdicu,listmmpfndi){
  flistmmpfndicu<-sample(listmmpfndi,size=round(length(listmmpfndi)*pfdi
cu))
  return(flistmmpfndicu)
}
```

3.3. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物不使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfndicudom<-function(pfdicu,pmpfneddlicudom,listmmpfndi){
  listmmpfndicu<-listmmpfndicu(pfdicu,listmmpfndi)
  fpigmmpfndicudom<-rbinom(length(listmmpfndicu),
                             mfarm[listmmpfndicu]*litter,
                             pmpfneddlicudom*pmaint)
  return(fpigmmpfndicudom)
}
```

4. 細菌性下痢症が発生したがコリスチンによる治療が実施されなかった農場

4.1. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfndicn<-function(pfdicu,listmmpfndi){
  flistmmpfndicn<-sample(listmmpfndi,
                          size=round(length(listmmpfndi)*(1-pfdicu)))
  return(flistmmpfndicn)
}
```

4.2. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfndicndom<-function(pfdicu,pmpfneddlicndom,listmmpfndi){
  listmmpfndicn<-listmmpfndicn(pfdicu,listmmpfndi)
```

```
fpigmmpfndicndom<-rbinom(length(lstmmpfndicn),  
                           mfarm[lstmmpfndicn]*litter,  
                           pmpfneddicndom)  
return(fpigmmpfndicndom)  
}
```

5. 浮腫病と細菌性下痢症の両方が発生しコリスチンによる治療が行われた農場

5.1. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物
不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfneddicu<-function(pmffndi,lstmmpfnedcu){  
  flistmmpfneddicu<-sample(lstmmpfnedcu,  
                             size=round(length(lstmmpfnedcu)*pmffndi))  
  return(flistmmpfneddicu)  
}
```

5.2. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物
不使用中規模 *mcr* 侵入農場で、上記 1 と 3 でダブルカウントしている出荷時 *mcr* 保有コ
リスチン耐性大腸菌優勢豚の頭数

```
pigmmpfncuoverlap<-function(pmffndi,pmpfneddicudom,lstmmpfnedcu){  
  lstmmpfneddicu<-listmmpfneddicu(pmffndi,lstmmpfnedcu)  
  fpigmmpfncuoverlap<-rbinom(length(lstmmpfneddicu),  
                               mfarm[lstmmpfneddicu]*litter,  
                               pmpfneddicudom*pmaint)  
  return(fpigmmpfncuoverlap)  
}
```


6. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が実施されなかった農場

6.1. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場のリスト (ID 番号)

```
listmmpfneddicn<-function(pmffndi,lstmmpfnedcn){  
  flstmmpfneddicn<-sample(lstmmpfnedcn,  
                           size=round(length(lstmmpfnedcn)*pmffndi))  
  return(flstmmpfneddicn)  
}
```

6.2. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfncnoverlap<-function(pmffndi,pmpfneddicndom,lstmmpfnedcn){  
  lstmmpfneddicn<-listmmpfneddicn(pmffndi,lstmmpfnedcn)  
  fpigmmpfncnoverlap<-rbinom(length(lstmmpfneddicn),  
                              mfarm[lstmmpfneddicn]*litter,  
                              pmpfneddicndom)  
  return(fpigmmpfncnoverlap)  
}
```

7. 浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった農場

浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった飼料添加物不使用中規模 *mcr* 侵入農場で、出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
pigmmpfnnoeddicndom<-function(pmffned,pmffndi,pmpfnnoeddicndom,lstmmpfn){  
  fpigmmpfnnoeddicndom<-round(sum(mfarm[lstmmpfn]*litter)*  
                               (1-(pmffned+pmffndi-pmffned*pmffndi))*pmpfnnoeddicndom)
```

```
# fpigmmpfnnoeddicndom<-rbinom(round(length(listmmpfn(nmmp,pfa))*max
(0.001,(1-(pmffned+pmffndi-pmffned*pmffndi)))),
#                                     mfarm[listmmpfn(nmmp,pfa)]*litter,
#                                     pmpfnnoeddicndom)
return(fpigmmpfnnoeddicndom)
}
```

8. 合計頭数

8.1. 飼料添加物不使用中規模 *mcr* 侵入農場で生産され出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の合計頭数

```
totalmcrpigmmpfn<-function(pmffned,pmffndi,pfdicu,pmped,
                           pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,
                           lstmmpfn,lstmmpfnedcu,lstmmpfnedcn){
  lstmmpfndi<-listmmpfndi(pmffndi,lstmmpfn)
  ftotalmcrpigmmpfn<-sum(pigmmpfneddicudom(pmped,pmpfneddicudom,lstmmpfn
edcu))+
  sum(pigmmpfneddicndom(pmped,pmpfneddicndom,lstmmpfnedcn))+
  sum(pigmmpfneddicudom(pfdicu,pmpfneddicudom,lstmmpfndi))+
  sum(pigmmpfneddicndom(pfdicu,pmpfneddicndom,lstmmpfndi))-
  sum(pigmmpfneddicuoverlap(pmffndi,pmpfneddicudom,lstmmpfnedcu))-
  sum(pigmmpfneddicuoverlap(pmffndi,pmpfneddicndom,lstmmpfnedcn))+
  pigmmpfnnoeddicndom(pmffned,pmffndi,pmpfnnoeddicndom,lstmmpfn)
  return(ftotalmcrpigmmpfn)
}
```

8.2. 飼料添加物不使用中規模 *mcr* 侵入農場における浮腫病による合計死亡頭数

```
totaldeathmmpfn<-function(pmped,lstmmpfnedcu,lstmmpfnedcn){
  ftotaldeathmmpfn<-sum(deathmmpfnedcu(pmped,lstmmpfnedcu))+
  sum(deathmmpfnedcn(pmped,lstmmpfnedcn))
  return(ftotaldeathmmpfn)
}
```


8.3. 飼料添加物不使用中規模 *mcr* 侵入農場で生産された健康豚の合計頭数

(*pmcrpiglmpfn* と *integratedslpigmp* に使用されているが、リスク推定には直接使用されていない)

```
totalslmpfn<-function(pfedcu,pmped,lstmpfn,lstmpfned){  
  lstmpfnedcu<-listmpfnedcu(pfedcu,lstmpfned)  
  # Lstmpfnedcn<-listmpfnedcn(pfedcu,lstmpfned)  
  lstmpfnedcn<-setdiff(lstmpfned,lstmpfnedcu)  
  ftotalslmpfn<-sum(mfarm[lstmpfn]*litter)-  
  totaldeathmpfn(pmped,lstmpfnedcu,lstmpfnedcn)  
  return(ftotalslmpfn)  
}
```

9. このグループのリスク

出荷時の *mcr* 保有コリスチン耐性大腸菌優勢豚割合（詳細なモニタリング用。モデルには未使用。）

```
pmcrpigmpfn<-function(nmmp,pmffned,pmffndi,pfedcu,pfdicu,pmped,  
  pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom){  
  lstmp<-listmp(nmmp)  
  lstmpfn<-listmpfn(pfa,lstmp)  
  lstmpfned<-listmpfned(pmffned,lstmpfn)  
  lstmpfnedcu<-listmpfnedcu(pfedcu,lstmpfned)  
  # Lstmpfnedcn<-listmpfnedcn(pfedcu,lstmpfned)  
  lstmpfnedcn<-setdiff(lstmpfned,lstmpfnedcu)  
  fpmcrpigmpfn<-totalmcrpigmpfn(pmffned,pmffndi,pfdicu,pmped,  
    pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,lstmpfn,lstmpfnedcu,lstmpfnedcn)/  
  totalslmpfn(pfedcu,pmped,lstmpfn,lstmpfned)  
  return(fpmcrpigmpfn)  
}
```

コリスチン リスクモデル コード「飼料添加物使用大規模 *mcr* 侵入農場」

ワークシート「Lf_mcrp_feedu.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、飼料添加物使用大規模 *mcr* 侵入農場における耐性選択のダイナミクスを表現するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Lf_mcrp_feedu.R」という名前で保存し、「Shift」+「Ctrl」+「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一字一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

飼料添加物使用大規模 *mcr* 侵入農場のリスト作成

```
listlmpfu<-function(pfa,lstlmp){  
  flistlmpfu<-sample(lstlmp,size=round(length(lstlmp)*pfa))  
  return(flistlmpfu)  
}
```

飼料添加物使用大規模 *mcr* 侵入農場において母豚から生まれた子豚の総数

(注：モデルには不使用)

```
piglmpfu<-function(lstlmpfu){  
  fpiglmpfu<-lfarm[lstlmpfu]*litter  
  return(fpiglmpfu)  
}
```


1. 浮腫病が発生し、コリスチンによる治療が行われた農場

1.1. 浮腫病の発生した飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfued<-function(plffued,lstlmpfu){  
  flistlmpfued<-sample(lstlmpfu,size=round(length(lstlmpfu)*plffued))  
  return(flistlmpfued)  
}
```

1.2. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfuedcu<-function(pfedcu,lstlmpfued){  
  flistlmpfuedcu<-sample(lstlmpfued,  
                          size=round(length(lstlmpfued)*pfedcu))  
  return(flistlmpfuedcu)  
}
```

1.3. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用大規模 *mcr* 侵入農場で死んだ豚の頭数（発症豚は全頭死亡）

```
deathlmpfuedcu<-function(plped,lstlmpfuedcu){  
  fdeathlmpfuedcu<-rbinom(length(lstlmpfuedcu),  
                           lfarm[lstlmpfuedcu]*litter,plped)  
  return(fdeathlmpfuedcu)  
}
```

1.4. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfuedcudom<-function(plped,pmpfueddicudom,lstlmpfuedcu){  
  fpiglmpfuedcudom<-rbinom(length(lstlmpfuedcu),  
                             lfarm[lstlmpfuedcu]*litter,  
                             (1-plped)*pmpfueddicudom*pmaint)  
  return(fpiglmpfuedcudom)  
}
```

2. 浮腫病が発生したがコリスチンによる治療をしない農場

2.1. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfuedcn<-function(pfedcu,lstlmpfued){  
  flistlmpfuedcn<-sample(lstlmpfued,  
                           size=round(length(lstlmpfued)*(1-pfedcu)))  
  return(flistlmpfuedcn)  
}
```

2.2. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場で死亡した豚の頭数

```
deathlmpfuedcn<-function(plped,lstlmpfuedcn){  
  fdeathlmpfuedcn<-rbinom(length(lstlmpfuedcn),  
                            lfarm[lstlmpfuedcn]*litter,plped)  
  return(fdeathlmpfuedcn)  
}
```

2.3. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfuedcndom<-function(plped,pmpfueddicndom,lstlmpfuedcn){  
  fpiglmpfuedcndom<-rbinom(length(lstlmpfuedcn),  
                             lfarm[lstlmpfuedcn]*litter,  
                             (1-plped)*pmpfueddicndom)  
  return(fpiglmpfuedcndom)  
}
```

3. 細菌性下痢症が発生しコリスチンによる治療が実施された農場

3.1. 下痢症の発生した飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfudi<-function(plffudi,lstlmpfu){  
  flistlmpfudi<-sample(lstlmpfu,size=round(length(lstlmpfu)*plffudi))
```



```
    return(flistlmpfudi)
}
```

3.2. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfudicu<-function(pfdicu,lstlmpfudi){
  flistlmpfudicu<-sample(lstlmpfudi,size=round(length(lstlmpfudi)*pfdi
cu))
  return(flistlmpfudicu)
}
```

3.3. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfudicudom<-function(pfdicu,pmpfueddicudom,piglmpfudi){
  lstlmpfudicu<-listlmpfudicu(pfdicu,piglmpfudi)
  fpiglmpfudicudom<-rbinom(length(lstlmpfudicu),
                             lfarm[lstlmpfudicu]*litter,
                             pmpfueddicudom*pmaint)
  return(fpiglmpfudicudom)
}
```

4. 細菌性下痢症が発生したがコリスチンによる治療が実施されなかった農場

4.1. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfudicn<-function(pfdicu,lstlmpfudi){
  flistlmpfudicn<-sample(lstlmpfudi,
                           size=round(length(lstlmpfudi)*(1-pfdicu)))
  return(flistlmpfudicn)
}
```

4.2. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfudicndom<-function(pfdicu,pmpfueddicndom,lstlmpfudi){
  lstlmpfudicn<-listlmpfudicn(pfdicu,lstlmpfudi)
}
```

```
fpiglmpfudicndom<-rbinom(length(lstlmpfudicn),  
                           lfarm[lstlmpfudicn]*litter,  
                           pmpfueddicndom)  
return(fpiglmpfudicndom)  
}
```

5. 浮腫病と細菌性下痢症の両方が発生しコリスチンによる治療が行われた農場

5.1. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物
使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfueddicu<-function(plffudi,lstlmpfuedcu){  
  flistlmpfueddicu<-sample(lstlmpfuedcu,  
                            size=round(length(lstlmpfuedcu)*plffudi))  
  return(flistlmpfueddicu)  
}
```

5.2. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物
使用大規模 *mcr* 侵入農場で、上記 1 と 3 でダブルカウントしている出荷時 *mcr* 保有コリス
チン耐性大腸菌優勢豚の頭数

```
piglmpfucuoverlap<-function(plffudi,pmpfueddicudom,lstlmpfuedcu){  
  lstlmpfueddicu<-listlmpfueddicu(plffudi,lstlmpfuedcu)  
  fpiglmpfucuoverlap<-rbinom(length(lstlmpfueddicu),  
                              lfarm[lstlmpfueddicu]*litter,  
                              pmpfueddicudom*pmain)  
  return(fpiglmpfucuoverlap)  
}
```


6. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が実施され

なかった農場

6.1. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfueddicn<-function(plffudi,lstlmpfuedcn){  
  flistlmpfueddicn<-sample(lstlmpfuedcn,  
                             size=round(length(lstlmpfuedcn)*plffudi))  
  return(flistlmpfueddicn)  
}
```

6.2. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfucoverlap<-function(plffudi,pmpfueddicndom,lstlmpfuedcn){  
  lstlmpfueddicn<-listlmpfueddicn(plffudi,lstlmpfuedcn)  
  fpiglmpfucoverlap<-rbinom(length(lstlmpfueddicn),  
                              lfarm[lstlmpfueddicn]*litter,  
                              pmpfueddicndom)  
  return(fpiglmpfucoverlap)  
}
```

7. 浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった農場

浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった飼料添加物使用大規模 *mcr* 侵入農場で、出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfunoeddicndom<-function(plffued,plffudi,pmpfunoeddicndom,lstlmpfu){  
  fpiglmpfunoeddicndom<-round(sum(lfarm[lstlmpfu]*litter)*  
                               (1-(plffued+plffudi-plffued*plffudi))*pmpfunoeddicndom)  
  # fpiglmpfunoeddicndom<-rbinom(1,(sum(lfarm[lstlmpfu(nlmp,pfa)]*litt
```

```
er))*(1-(plffued+plffudi-plffued*plffudi)),
#                                pmpfunoeddicndom)
  return(fpiglmpfunoeddicndom)
}
```

8. 合計頭数

8.1. 飼料添加物使用大規模 *mcr* 侵入農場で生産され出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の合計頭数

```
totalmcrpiglmpfu<-function(plffued,plffudi,pfdicu,plped,
                             pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,
                             lstlmpfu,lstlmpfuedcu,lstlmpfuedcn){
  lstlmpfudi<-listlmpfudi(plffudi,lstlmpfu)
  ftotalmcrpiglmpfu<-sum(piglmpfuedcudom(plped,pmpfueddicudom,lstlmpfuedcu))+
  sum(piglmpfuedcndom(plped,pmpfueddicndom,lstlmpfuedcn))+
  sum(piglmpfudicudom(pfdicu,pmpfueddicudom,lstlmpfudi))+
  sum(piglmpfudicndom(pfdicu,pmpfueddicndom,lstlmpfudi))-
  sum(piglmpfucuooverlap(plffudi,pmpfueddicudom,lstlmpfuedcu))-
  sum(piglmpfucnooverlap(plffudi,pmpfueddicndom,lstlmpfuedcn))+
  piglmpfunoeddicndom(plffued,plffudi,pmpfunoeddicndom,lstlmpfu)
  return(ftotalmcrpiglmpfu)
}
```

8.2. 飼料添加物使用大規模 *mcr* 侵入農場における浮腫病による合計死亡頭数

```
totaldeathlmpfu<-function(plped,lstlmpfuedcu,lstlmpfuedcn){
  ftotaldeathlmpfu<-sum(deathlmpfuedcu(plped,lstlmpfuedcu))+
  sum(deathlmpfuedcn(plped,lstlmpfuedcn))
  return(ftotaldeathlmpfu)
}
```

8.3. 飼料添加物使用大規模 *mcr* 侵入農場で生産された健康豚の合計頭数（*pmcrlmpfu* と *integratedslpigmp* に使用されているが、リスク推定には直接使用されていない）


```
totalsllmpfu<-function(pfedcu,plped,lstlmpfu,lstlmpfued){  
  lstlmpfuedcu<-listlmpfuedcu(pfedcu,lstlmpfued)  
  # Lstlmpfuedcn<-listlmpfuedcn(pfedcu,lstlmpfued)  
  lstlmpfuedcn<-setdiff(lstlmpfued,lstlmpfuedcu)  
  ftotalsllmpfu<-sum(lfarm[lstlmpfu]*litter)-  
  totaldeathlmpfu(plped,lstlmpfuedcu,lstlmpfuedcn)  
  return(ftotalsllmpfu)  
}
```

9. このグループのリスク

出荷時の *mcr* 保有コリスチン耐性大腸菌優勢豚割合（詳細なモニタリング用。モデルには未使用。）

```
pmcrpiglmpfu<-function(nlmp,plffued,plffudi,pfedcu,pfdicu,plped,  
  pmpfneddlicudom,pmpfueddicndom,pmpfunoeddlicndom){  
  lstlmp<-listlmp(nlmp)  
  lstlmpfu<-listlmpfu(pfa,lstlmp)  
  lstlmpfued<-listlmpfued(plffued,lstlmpfu)  
  lstlmpfuedcu<-listlmpfuedcu(pfedcu,lstlmpfued)  
  # Lstlmpfuedcn<-listlmpfuedcn(pfedcu,lstlmpfued)  
  lstlmpfuedcn<-setdiff(lstlmpfued,lstlmpfuedcu)  
  fpmcrpiglmpfu<-totalmcrpiglmpfu(plffued,plffudi,pfdicu,plped,  
    pmpfneddlicudom,pmpfueddicndom,pmpfunoeddlicndom,  
    lstlmpfu,lstlmpfuedcu,lstlmpfuedcn)/  
  totalsllmpfu(pfedcu,plped,lstlmpfu,lstlmpfued)  
  return(fpmcrpiglmpfu)  
}
```

コリスチン リスクモデル コード「飼料添加物不使用大規模 *mcr* 侵入農場」

ワークシート「Lf_mcrp_feedn.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、飼料添加物不使用大規模 *mcr* 侵入農場における耐性選択のダイナミクスを表現するものです。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Lf_mcrp_feedn.R」という名前で保存し、「Shift」+「Ctrl」+「Enter」で全行を読ませてください。あるいはもし本文章全体を張り付ける場合は、コード以外の行の一字一文字目に # を置くことで、R が読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

飼料添加物不使用大規模 *mcr* 侵入農場のリスト作成

2017 年度モデルの時に該当数が 0 になるとエラーが起こる可能性があるために少なくとも 1 農場が選ばれるようセットしてある。

```
listlmpfn<-function(pfa,lstlmp){  
  # flistlmpfn<-sample(lstlmp,size=round(length(lstlmp)*(1-pfa)))  
  flistlmpfn<-sample(lstlmp,size=max(1,round(length(lstlmp)*(1-pfa))))  
  return(flistlmpfn)  
}
```

飼料添加物不使用大規模 *mcr* 侵入農場において母豚から生まれた子豚の総数

(注：モデルには不使用)

```
piglmpfn<-function(lstlmp){  
  fpiglmpfn<-lfarm[lstlmp]*litter  
  return(fpiglmpfn)  
}
```


1. 浮腫病が発生し、コリスチンによる治療が行われた農場

1.1. 浮腫病の発生した飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfned<-function(plffned,lstlmpfn){  
  flistlmpfned<-sample(lstlmpfn,size=round(length(lstlmpfn)*plffned))  
  return(flistlmpfned)  
}
```

1.2. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfnedcu<-function(pfedcu,lstlmpfned){  
  flistlmpfnedcu<-sample(lstlmpfned,  
                           size=round(length(lstlmpfned)*pfedcu))  
  return(flistlmpfnedcu)  
}
```

1.3. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用大規模 *mcr* 侵入農場で死んだ豚の頭数（発症豚は全頭死亡）

```
deathlmpfnedcu<-function(plped,lstlmpfnedcu){  
  fdeathlmpfnedcu<-rbinom(length(lstlmpfnedcu),  
                            lfarm[lstlmpfnedcu]*litter,plped)  
  return(fdeathlmpfnedcu)  
}
```

1.4. 浮腫病が発生し、コリスチンによる治療が行われた飼料添加物不使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfnedcudom<-function(plped,pmpfneddicudom,lstlmpfnedcu){  
  fpiglmpfnedcudom<-rbinom(length(lstlmpfnedcu),  
                             lfarm[lstlmpfnedcu]*litter,  
                             (1-plped)*pmpfneddicudom*pmaint)  
  return(fpiglmpfnedcudom)  
}
```

2. 浮腫病が発生したがコリスチンによる治療をしない農場

2.1. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfnedcn<-function(pfedcu,lstlmpfned){  
  flistlmpfnedcn<-sample(lstlmpfned,  
                           size=round(length(lstlmpfned)*(1-pfedcu)))  
  return(flistlmpfnedcn)  
}
```

2.2. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場で死亡した豚の頭数

```
deathlmpfnedcn<-function(plped,lstlmpfnedcn){  
  fdeathlmpfnedcn<-rbinom(length(lstlmpfnedcn),  
                             lfarm[lstlmpfnedcn]*litter,plped)  
  return(fdeathlmpfnedcn)  
}
```

2.3. 浮腫病が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfnedcndom<-function(plped,pmpfneddicndom,lstlmpfnedcn){  
  fpiglmpfnedcndom<-rbinom(length(lstlmpfnedcn),  
                              lfarm[lstlmpfnedcn]*litter,  
                              (1-plped)*pmpfneddicndom)  
  return(fpiglmpfnedcndom)  
}
```

3. 細菌性下痢症が発生しコリスチンによる治療が実施された農場

3.1. 下痢症の発生した飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfndi<-function(plffndi,lstlmpfn){  
  flistlmpfndi<-sample(lstlmpfn,size=round(length(lstlmpfn)*plffndi))
```



```
return(flistlmpfndi)
}
```

3.2. 下痢症が発生しコリスチンによる治療が行われた飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfndicu<-function(pfdicu,listlmpfndi){
  flistlmpfndicu<-sample(listlmpfndi,size=round(length(listlmpfndi)*pfdi
cu))
  return(flistlmpfndicu)
}
```

3.3. 下痢症が発生し、コリスチンによる治療が行われた飼料添加物不使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfndicudom<-function(pfdicu,pmpfneddlicudom,listlmpfndi){
  listlmpfndicu<-listlmpfndicu(pfdicu,listlmpfndi)
  fpiglmpfndicudom<-rbinom(length(listlmpfndicu),
                             lfarm[listlmpfndicu]*litter,
                             pmpfneddlicudom*pmaint)
  return(fpiglmpfndicudom)
}
```

4. 細菌性下痢症が発生したがコリスチンによる治療が実施されなかった農場

4.1. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfndicn<-function(pfdicu,listlmpfndi){
  flistlmpfndicn<-sample(listlmpfndi,
                          size=round(length(listlmpfndi)*(1-pfdicu)))
  return(flistlmpfndicn)
}
```

4.2. 下痢症が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfndicndom<-function(pfdicu,pmpfneddlicndom,listlmpfndi){
  listlmpfndicn<-listlmpfndicn(pfdicu,listlmpfndi)
```

```
fpiglmpfndicndom<-rbinom(length(lstlmpfndicn),  
                           lfarm[lstlmpfndicn]*litter,  
                           pmpfneddicndom)  
return(fpiglmpfndicndom)  
}
```

5. 浮腫病と細菌性下痢症の両方が発生しコリスチンによる治療が行われた農場

5.1. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物
不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfneddicu<-function(plffndi,lstlmpfnedcu){  
  flistlmpfneddicu<-sample(lstlmpfnedcu,  
                            size=round(length(lstlmpfnedcu)*plffndi))  
  return(flistlmpfneddicu)  
}
```

5.2. 浮腫病と細菌性下痢症の両方が発生し、コリスチンによる治療が行われた飼料添加物
不使用大規模 *mcr* 侵入農場で、上記 1 と 3 でダブルカウントしている出荷時 *mcr* 保有コリ
スチン耐性大腸菌優勢豚の頭数

```
piglmpfncuoverlap<-function(plffndi,pmpfneddicudom,lstlmpfnedcu){  
  lstlmpfneddicu<-listlmpfneddicu(plffndi,lstlmpfnedcu)  
  fpiglmpfncuoverlap<-rbinom(length(lstlmpfneddicu),  
                              lfarm[lstlmpfneddicu]*litter,  
                              pmpfneddicudom*pmaint)  
  return(fpiglmpfncuoverlap)  
}
```


6. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が実施され

なかった農場

6.1. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場のリスト (ID 番号)

```
listlmpfneddicn<-function(plffndi,lstlmpfnedcn){  
  flistlmpfneddicn<-sample(lstlmpfnedcn,  
                             size=round(length(lstlmpfnedcn)*plffndi))  
  return(flistlmpfneddicn)  
}
```

6.2. 浮腫病と細菌性下痢症の両方が発生したがコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場で出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfncnoverlap<-function(plffndi,pmpfneddicndom,lstlmpfnedcn){  
  lstlmpfneddicn<-listlmpfneddicn(plffndi,lstlmpfnedcn)  
  fpiglmpfncnoverlap<-rbinom(length(lstlmpfneddicn),  
                               lfarm[lstlmpfneddicn]*litter,  
                               pmpfneddicndom)  
  return(fpiglmpfncnoverlap)  
}
```

7. 浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった農場

浮腫病も下痢症も発生なくコリスチンによる治療が行われなかった飼料添加物不使用大規模 *mcr* 侵入農場で、出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の頭数

```
piglmpfnnoeddicndom<-function(plffned,plffndi,pmpfnnoeddicndom,lstlmpfn){  
  fpiglmpfnnoeddicndom<-round(sum(lfarm[lstlmpfn]*litter)*  
                               (1-(plffned+plffndi-plffned*plffndi))  
                               *pmpfnnoeddicndom)
```

```
# fpiglmpfnnoeddicndom<-rbinom(round(length(listlmpfn(nlmp,pfa))*ma
x(0.001,(1-(plffned+plffndi-plffned*plffndi)))),
#                               lfarm[listlmpfn(nlmp,pfa)]*litter,
#                               pmpfnnoeddicndom)
return(fpiglmpfnnoeddicndom)
}
```

8. 合計頭数

8.1. 飼料添加物不使用大規模 *mcr* 侵入農場で生産され出荷直前に *mcr* 保有コリスチン耐性大腸菌が優勢である豚の合計頭数

```
totalmcrpiglmpfn<-function(plffned,plffndi,pfdicu,plped,
                           pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,
                           lstlmpfn,lstlmpfnedcu,lstlmpfnedcn){
  lstlmpfndi<-listlmpfndi(plffndi,lstlmpfn)
  ftotalmcrpiglmpfn<-sum(piglmpfneddicudom(plped,pmpfneddicudom,lstlmpfn
edcu))+
  sum(piglmpfneddicndom(plped,pmpfneddicndom,lstlmpfnedcn))+
  sum(piglmpfneddicudom(pfdicu,pmpfneddicudom,lstlmpfndi))+
  sum(piglmpfneddicndom(pfdicu,pmpfneddicndom,lstlmpfndi))-
  sum(piglmpfnedcuoverlap(plffndi,pmpfneddicudom,lstlmpfnedcu))-
  sum(piglmpfnedcnoverlap(plffndi,pmpfneddicndom,lstlmpfnedcn))+
# piglmpfnnoeddicndom(nlmp,pfa,plffned,plffndi)
  piglmpfnnoeddicndom(plffned,plffndi,pmpfnnoeddicndom,lstlmpfn)
  return(ftotalmcrpiglmpfn)
}
```

8.2. 飼料添加物不使用大規模 *mcr* 侵入農場における浮腫病による合計死亡頭数

```
totaldeathlmpfn<-function(plped,lstlmpfnedcu,lstlmpfnedcn){
  ftotaldeathlmpfn<-sum(deathlmpfnedcu(plped,lstlmpfnedcu))+
  sum(deathlmpfnedcn(plped,lstlmpfnedcn))
  return(ftotaldeathlmpfn)
}
```


8.3. 飼料添加物不使用大規模 *mcr* 侵入農場で生産された健康豚の合計頭数

(*pmcrpiglmpfn* と *integratedslpigmp* に使用されているが、リスク推定には直接使用されていない)

```
totalsllmpfn<-function(pfedcu,plped,lstlmpfn,lstlmpfnd){  
  lstlmpfndcu<-listlmpfndcu(pfedcu,lstlmpfnd)  
  # lstlmpfndcn<-listlmpfndcn(pfedcu,lstlmpfnd)  
  lstlmpfndcn<-setdiff(lstlmpfnd,lstlmpfndcu)  
  ftotalsllmpfn<-sum(lfarm[lstlmpfn]*litter)-  
  totaldeathlmpfn(plped,lstlmpfndcu,lstlmpfndcn)  
  return(ftotalsllmpfn)  
}
```

9. このグループのリスク

出荷時の *mcr* 保有コリスチン耐性大腸菌優勢豚割合（詳細なモニタリング用。モデルには未使用。）

```
pmcrpiglmpfn<-function(nlmp,pfa,plffned,plffndi,pfedcu,pfdicu,plped,  
  pmpfneddlicudom,pmpfneddlicndom,pmpfnnoeddlicndom){  
  lstlmp<-listlmp(nlmp)  
  lstlmpfn<-listlmpfn(pfa,lstlmp)  
  lstlmpfnd<-listlmpfnd(plffned,lstlmpfn)  
  lstlmpfndcu<-listlmpfndcu(pfedcu,lstlmpfnd)  
  # lstlmpfndcn<-listlmpfndcn(pfedcu,lstlmpfnd)  
  lstlmpfndcn<-setdiff(lstlmpfnd,lstlmpfndcu)  
  fpmcrpiglmpfn<-totalmcrpiglmpfn(plffned,plffndi,pfdicu,plped,  
    pmpfneddlicudom,pmpfneddlicndom,pmpfnnoeddlicndom,  
    lstlmpfn,lstlmpfndcu,lstlmpfndcn)/  
  totalsllmpfn(pfedcu,plped,lstlmpfn,lstlmpfnd)  
  return(fpmcrpiglmpfn)  
}
```

コリスチン リスクモデル コード「モデルシート」

ワークシート「Modelsheet.R」に記載したコードの説明

以下のコードは、プラスミド性コリスチン耐性大腸菌腸管内優勢豚の発生評価に必要な、モデルを走らせる心臓部分です。最初に農場設定に必要なパラメータを指定する部分があり、その後他のワークシートを統合してモデルを走らせ、結果を表示させるコードが書かれています。ご使用の際には RStudio で新しいワークシートに本説明書のコード部分のみを張り付け、「Modelsheet.R」という名前で保存し、まず予めパッケージ「tcltk」をインストールし、このライブラリを起動させ、他のワークシートを「Shift」＋「Ctrl」＋「Enter」で順番に全て読み込ませます。次に、「Modelsheet.R」の一番上のパラメータ設定部分をハイライトし、「Ctrl」＋「Enter」で該当部分を読ませます。最後に、source()からモデルを走らせる「Bintegratedrisk」までの部分をハイライトし、「Ctrl」＋「Enter」で該当部分を読ませてください。各飼養規模のリスク評価をするには、統合モデルの後の部分を同様に走らせます。もし本文章全体を張り付ける場合は、コード以外の行の一文字目に#を置くことで、Rが読まないよう処理することが出来ます。本説明書は、R Markdown を使用して、RStudio から出力して作成しています。

パラメータ値の設定

小規模農場数、小規模農場の母豚数下限と上限

```
n_sfarms <- 212  
n_spig_lower<-10  
n_spig_upper<-50
```

中規模農場数、中規模農場の母豚数下限と上限

```
n_mfarms <- 474  
n_mpig_lower<-51  
n_mpig_upper<-200
```

大規模農場数、大規模農場の母豚数下限と上限


```
n_lfarms <- 314  
n_lpig_lower<-201  
n_lpig_upper<-600
```

母豚当たり産仔数

```
litter<-12
```

統合モデル

ワークシートの呼び出し

```
source("Probabilities.R")  
source("Farm_pig.R")  
source("Sf_mcrp_feedu.R")  
source("Sf_mcrp_feedn.R")  
source("Mf_mcrp_feedu.R")  
source("Mf_mcrp_feedn.R")  
source("Lf_mcrp_feedu.R")  
source("Lf_mcrp_feedn.R")
```

出荷直前で *mcr* 保有コリスチン耐性大腸菌が優勢となっている豚の合計頭数

各農場カテゴリーのワークシートから合計頭数を読み込んでいる。

```
integratedmcrpig<-function(  
  plffued,plffned,pmffued,pmffned,psffued,psffned,  
  plffudi,plffndi,pmffudi,pmffndi,psffudi,psffndi,  
  pfdicu,plped,pmped,psped,  
  pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
  pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,  
  lstlmpfu,lstlmpfn,lstmpmfu,lstmpfn,lstmpfu,
```

```
lstmpfn,lstmpfuedcu,lstmpfuedcu,lstmpfuedcu,  
lstmpfuedcu,lstmpfuedcu,lstmpfuedcu,lstmpfued  
cn,  
  
lstmpfuedcn,lstmpfuedcn,lstmpfuedcn,lstmpfued  
cn,  
  
lstmpfuedcn){  
fintmcrpig<-totalmcrpiglmpfu(plffued,plffudi,pfdicu,plped,  
pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
  
lstlmpfu,lstlmpfuedcu,lstlmpfuedcn)+  
totalmcrpiglmpfn(plffned,plffndi,pfdicu,plped,  
pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,  
  
lstlmpfn,lstlmpfuedcu,lstlmpfuedcn)+  
totalmcrpigmpfu(pmffued,pmffudi,pfdicu,pmped,  
pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
  
lstmmpfu,lstmmpfuedcu,lstmmpfuedcn)+  
totalmcrpigmpfn(pmffned,pmffndi,pfdicu,pmped,  
pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,  
  
lstmmpfn,lstmmpfuedcu,lstmmpfuedcn)+  
totalmcrpigsmpfu(psffued,psffudi,pfdicu,psped,  
pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
  
lstmpfu,lstmpfuedcu,lstmpfuedcn)+  
totalmcrpigsmpfn(psffned,psffndi,pfdicu,psped,  
pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,  
  
lstmpfn,lstmpfuedcu,lstmpfuedcn)  
return(fintmcrpig)  
}
```


*mcr*侵入農場で浮腫病により死亡した豚の合計頭数

```
totaldeathmp<-function(  
    plped,pmped,psped,  
    lstlmpfu,lstlmpfn,lstmpfu,lstmpfn,lstmpfu,  
    lstmpfn,lstlmpfuedcu,lstlmpfuedcu,lstmpfuedc  
u,  
    lstmpfuedcu,lstmpfuedcu,lstmpfuedcu,  
    lstlmpfuedcn,lstlmpfuedcn,lstmpfuedcn,  
    lstmpfuedcn,lstmpfuedcn,lstmpfuedcn){  
ftotaldeathmp<-totaldeathlmpfu(plped,lstlmpfuedcu,lstlmpfuedcn)+  
    totaldeathlmpfn(plped,lstlmpfuedcu,lstlmpfuedcn)+  
    totaldeathmpfu(pmped,lstmpfuedcu,lstmpfuedcn)+  
    totaldeathmpfn(pmped,lstmpfuedcu,lstmpfuedcn)+  
    totaldeathsmfu(psped,lstsmfuedcu,lstsmfuedcn)+  
    totaldeathsmfn(psped,lstsmfuedcu,lstsmfuedcn)  
return(ftotaldeathmp)  
}
```

作出した全 1000 農場で浮腫病により死亡した豚の合計頭数

```
totaldeath<-function(  
    plped,pmped,psped,  
    lstlmpfu,lstlmpfn,lstmpfu,lstmpfn,lstmpfu,  
    lstmpfn,lstlmpfuedcu,lstlmpfuedcu,lstmpfuedcu,  
    lstmpfuedcu,lstmpfuedcu,lstmpfuedcu,  
    lstlmpfuedcn,lstlmpfuedcn,lstmpfuedcn,lstmpfuedc  
n,  
    lstmpfuedcn,lstmpfuedcn){  
ftotaldeath<-totaldeathmp(  
    plped,pmped,psped,
```

```

    lstlmpfu, lstlmpfn, lstmmpfu, lstmmpfn, lstsmpf
u,

    lstsmpfn, lstlmpfuedcu, lstlmpfnedcu,

    lstmmpfuedcu, lstmmpfnedcu, lstsmpfuedcu,

    lstsmpfnedcu, lstlmpfuedcn, lstlmpfnedcn,

    lstmmpfuedcn, lstmmpfnedcn, lstsmpfuedcn,

    lstsmpfnedcn)*
  (sum(lfarm)+sum(mfarm)+sum(sfarm))/
  (sum(lfarm[lstlmpfu])+sum(lfarm[lstlmpfn])+
    sum(mfarm[lstmmpfu])+sum(mfarm[lstmmpfn])+
    sum(sfarm[lstsmpfu])+sum(sfarm[lstsmpfn]))
  return(ftotaldeath)
}

```

*mcr*侵入農場の出荷直前の健康豚合計頭数（モニタリングのためのコード。リス
 ク計算には使用していない。）

```

integratedslpigmp<-function(nlmp, nmmp, nsmp, pfa, plffued, plffned, pmffue
d, pmffned, psffued, psffned,
    pfedcu, plped, pmped, psped){
  lstlmp <- listlmp(nlmp)
  lstmmp <- listmmp(nmmp)
  lstsmp <- listsmp(nsmp)
  lstlmpfu <- listlmpfu(pfa, lstlmp)
  lstlmpfn <- setdiff(lstlmp, lstlmpfu)
  lstmmpfu <- listmmpfu(pfa, lstmmp)
  lstmmpfn <- setdiff(lstmmp, lstmmpfu)
  lstsmpfu <- listsmpfu(pfa, lstsmp)
  lstsmpfn <- setdiff(lstsmp, lstsmpfu)
  lstlmpfued <- listlmpfued(plffued, lstlmpfu)
  lstlmpfned <- listlmpfned(plffned, lstlmpfn)

```



```
lstmmpfued <- listmmpfued(pmffued,lstmmpfu)
lstmmpfned <- listmmpfned(pmffned,lstmmpfn)
lstsmpfued <- listsmpfued(psffued,lstsmpfu)
lstsmpfned <- listsmpfned(psffned,lstsmpfn)
fintslpig<-totalsllmpfu(pfedcu,plped,lstlmpfu,lstlmpfued)+
  totalsllmpfn(pfedcu,plped,lstlmpfn,lstlmpfned)+
  totalslmpfu(pfedcu,pmped,lstmmpfu,lstmmpfued)+
  totalslmpfn(pfedcu,pmped,lstmmpfn,lstmmpfned)+
  totalslmpfu(pfedcu,psped,lstsmpfu,lstsmpfued)+
  totalslmpfn(pfedcu,psped,lstsmpfn,lstsmpfned)
return(fintslpig)
}
```

1000 農場における出荷直前の豚における *mcr* 保有コリスチン耐性大腸菌優勢豚

の占める割合：リスク

```
integratedrisk<-function(nlmp,nmmp,nsmp,pfa,plffued,plffned,pmffued,
                          pmffned,psffued,psffned,
                          plffudi,plffndi,pmffudi,pmffndi,psffudi,psffnd
i,
                          pfedcu,pfdicu,plped,pmped,psped,
                          pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,
                          pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom)
{
  lstlmp <- listlmp(nlmp)
  lstmmp <- listmmp(nmmp)
  lstsmp <- listsmp(nsmp)
  lstlmpfu <- listlmpfu(pfa,lstlmp)
  lstlmpfn <- setdiff(lstlmp,lstlmpfu)
  lstmmpfu <- listmmpfu(pfa,lstmmp)
  lstmmpfn <- setdiff(lstmmp,lstmmpfu)
  lstsmpfu <- listsmpfu(pfa,lstsmp)
```

```
lstsmpfn <- setdiff(lstsmp, lstsmpfu)
lstlmpfued <- listlmpfued(plffued, lstlmpfu)
lstlmpfned <- listlmpfned(plffned, lstlmpfn)
lstmmpfued <- listmmpfued(pmffued, lstmmpfu)
lstmmpfned <- listmmpfned(pmffned, lstmmpfn)
lstsmpfued <- lstsmpfued(psffued, lstsmpfu)
lstsmpfned <- lstsmpfned(psffned, lstsmpfn)
lstlmpfuedcu <- listlmpfuedcu(pfedcu, lstlmpfued)
lstlmpfnedcu <- listlmpfnedcu(pfedcu, lstlmpfned)
lstmmpfuedcu <- listmmpfuedcu(pfedcu, lstmmpfued)
lstmmpfnedcu <- listmmpfnedcu(pfedcu, lstmmpfned)
lstsmpfuedcu <- lstsmpfuedcu(pfedcu, lstsmpfued)
lstsmpfnedcu <- lstsmpfnedcu(pfedcu, lstsmpfned)
lstlmpfuedcn <- setdiff(lstlmpfued, lstlmpfuedcu)
lstlmpfnedcn <- setdiff(lstlmpfned, lstlmpfnedcu)
lstmmpfuedcn <- setdiff(lstmmpfued, lstmmpfuedcu)
lstmmpfnedcn <- setdiff(lstmmpfned, lstmmpfnedcu)
lstsmpfuedcn <- setdiff(lstsmpfued, lstsmpfuedcu)
lstsmpfnedcn <- setdiff(lstsmpfned, lstsmpfnedcu)
fintrisk <- integratedmcrpig(plffued, plffned, pmffued, pmffned, psffued,
                             psffned,
                             plffudi, plffndi, pmffudi, pmffndi, psffudi, psffnd
i,
                             pfdicu, plped, pmped, psped,
                             pmpfueddicdom, pmpfueddicndom, pmpfunoeddicndom,
                             pmpfneddicdom, pmpfneddicndom, pmpfnnoeddicndom,
                             lstlmpfu, lstlmpfn, lstmmpfu, lstmmpfn, lstsmpfu,
                             lstsmpfn, lstlmpfuedcu, lstlmpfnedcu, lstmmpfue
dcu,
                             lstmmpfnedcu, lstsmpfuedcu, lstsmpfnedcu,
                             lstlmpfuedcn, lstlmpfnedcn, lstmmpfuedcn,
                             lstmmpfnedcn, lstsmpfuedcn, lstsmpfnedcn)/
((sum(lfarm)+sum(mfarm)+sum(sfarm))*litter-totaldeath(plped, pmped,
```



```

    psped,lstlmpfu,lstlmpfn,lstmpmfu,lstmpfn,
    lstsmpfu,lstsmpfn,lstlmpfuedcu,lstlmpfnedcu,
    lstmpmfuedcu,lstmpfnedcu,lstsmpfuedcu,
    lstsmpfnedcu,lstlmpfuedcn,lstlmpfnedcn,
    lstmpmfuedcn,lstmpfnedcn,lstsmpfuedcn,
    lstsmpfnedcn))
  return(fintrisk)
}

```

モデルを走らせるコード

統合モデル

プログレスバーでシミュレーションの進捗状況をモニタリングする。シミュレートされる 5000 回の結果を記憶させ、ファイルに格納する。

```

pb<-txtProgressBar(min = 0, max = length(nlmp), style = 3)

Bintegratedrisk <- numeric(length(nlmp))

for (i in seq_along(nlmp)) {
  Bintegratedrisk[i] <- integratedrisk(nlmp[i],nmmp[i],nsmp[i],pfa[i],
    plffued[i],plffned[i],pmffued[i],pmffned[i],
    psffued[i],psffned[i],
    plffudi[i],plffndi[i],pmffudi[i],pmffndi[i],
    psffudi[i],psffndi[i],
    pfedcu[i],pfdicu[i],plped[i],pmped[i],psped
    [i],
    pmpfueddicudom[i],pmpfueddicndom[i],pmpfunoe
    ddicndom[i],
    pmpfneddicudom[i],pmpfneddicndom[i],pmpfnnoe

```

```
ddicndom[i])  
  setTxtProgressBar(pb, i)  
}  
  
close(pb)  
  
save(Bintegratedrisk, file = "Bintegratedrisk.Rdata")
```

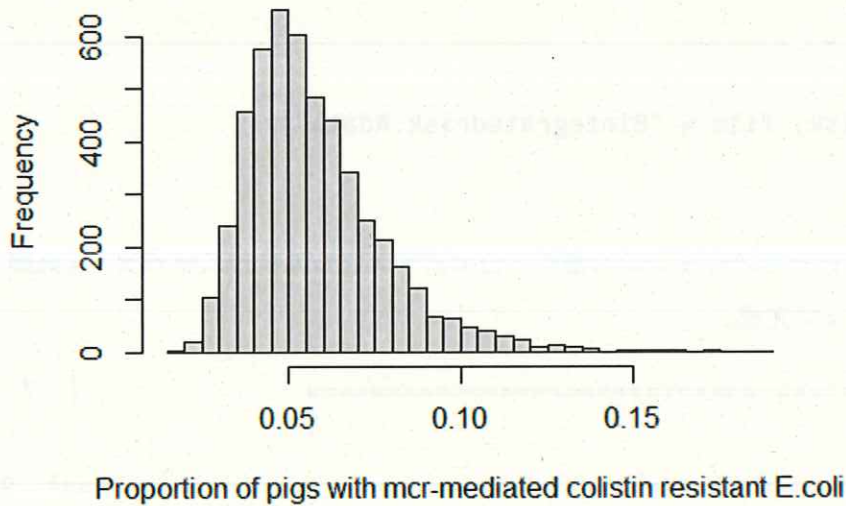
すると R コンソールにプログレスバーが現れ、以下のように左から右へ延びる。3 秒間
ほどでシミュレーション完成。

```
|=====| 7  
6% |  
|=====| 9  
9%  
|  
|=====| 10  
0%
```

シミュレーション結果を作図・表示する

```
windows(6,6)  
hist(Bintegratedrisk,breaks=40,xlab="Proportion of pigs with mcr-media  
ted colistin resistant E.coli",  
     col="grey80",main="")
```

外付けでウィンドウが立ち上がり、図が表示されるので、希望のフォーマットで保存する。



```
sortrisk<-sort(Bintegratedrisk)
```

```
mean(sortrisk)
```

```
## [1] 0.05761767
```

```
sortrisk[2500]
```

```
## [1] 0.05357623
```

```
sortrisk[0.25*5000]
```

```
## [1] 0.04375114
```

```
sortrisk[0.975*5000]
```

```
## [1] 0.107585
```

```
sortrisk[5000] # max
```

```
## [1] 0.18983
```

このように結果が表示される。平均値、中央値、2.5 および 97.5 パーセンタイル値を表

示。

各グループのリスク

1. 小規模農場

小規模農場における、出荷直前の豚に占める mcr 保有コリスチン耐性大腸菌優勢豚の割合。

```
intsrisk<-function(nsmf,pfa,psffued,psffned,psfffudi,psffndi,pfedcu,pfdicu,psped,
                    pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,
                    pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom)
{
  lstsmf <- listsmf(nsmf)
  lstsmfuf <- listsmfuf(pfa,lstsmf)
  lstsmfn <- setdiff(lstsmf,lstsmfuf)
  lstsmfued <- listsmfued(psffued,lstsmfuf)
  lstsmfned <- listsmfned(psffned,lstsmfn)
  lstsmfuedcu <- listsmfuedcu(pfedcu,lstsmfued)
  lstsmfnedcu <- listsmfnedcu(pfedcu,lstsmfned)
  lstsmfuedcn <- setdiff(lstsmfued,lstsmfuedcu)
  lstsmfnedcn <- setdiff(lstsmfned,lstsmfnedcu)
  fintsrisk<-
    (totalmcrpigsmpfu(psffued,psfffudi,pfdicu,psped,
                      pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,
                      lstsmfuf,lstsmfuedcu,lstsmfuedcn)+
    totalmcrpigsmpfn(psffned,psffndi,pfdicu,psped,
                      pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,
                      lstsmfn,lstsmfnedcu,lstsmfnedcn))/
  (sum(sfarm)*litter-
```



```
((totaldeathsmpfu(psped,lstsmpfuedcu,lstsmpfuedcn)+
totaldeathsmpfu(psped,lstsmpfuedcu,lstsmpfuedcn))*
sum(sfarm)/
(sum(sfarm[lstsmpfu])+sum(sfarm[lstsmpfu])))
))
return(fintsrisk)
}
```

小規模農場のモデルを走らせるコード プログレスバーでシミュレーションの進捗状況をモニタリングする。シミュレートされる 5000 回の結果を記憶させ、ファイルに格納する。

```
pbs<-txtProgressBar(min = 0, max = length(nsm), style = 3)

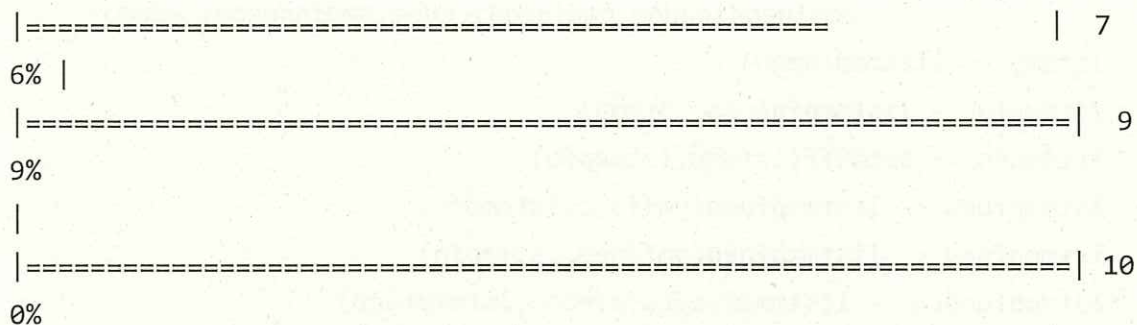
Bintsrisk <- numeric(length(nsm))

for (i in seq_along(nsm)) {
  Bintsrisk[i] <- intsrisk(nsm[i],pfa[i],
                           psffued[i],psffned[i],
                           psffudi[i],psffndi[i],
                           pfedcu[i],pfedcu[i],psped[i],
                           pmpfueddicudom[i],pmpfueddicndom[i],pmpfunoe
ddicndom[i],
                           pmpfneddicudom[i],pmpfneddicndom[i],pmpfnnoe
ddicndom[i])
  setTxtProgressBar(pbs, i)
}

close(pbs)

save(Bintsrisk, file = "Bintsrisk.Rdata")
```

すると R コンソールにプログレスバーが現れ、以下のように左から右へ延びる。3 秒間ほどでシミュレーション完成。



シミュレーション結果を表示する。

```
sortsrisk<-sort(Bintsrisk)
mean(sortsrisk)
## [1] 0.0483111

sortsrisk[2500]
## [1] 0.04461814

sortsrisk[0.25*5000]
## [1] 0.03505129

sortsrisk[0.975*5000]
## [1] 0.09534278

sortsrisk[5000]
## [1] 0.1701677
```

2. 中規模農場

```
intmrisk<-function(nmmp,pfa,pmffued,pmffned,pmffudi,pmffndi,pfedcu,
                    pfdicu,pmped,
                    pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,
```



```

      pmpfneddicudom, pmpfneddicndom, pmpfnnoeddicndom){
    lstmmp <- listmmp(nmmp)
    lstmmpfu <- listmmpfu(pfa, lstmmp)
    lstmmpfn <- setdiff(lstmmp, lstmmpfu)
    lstmmpfued <- listmmpfued(pmfffued, lstmmpfu)
    lstmmpfned <- listmmpfned(pmfffned, lstmmpfn)
    lstmmpfuedcu <- listmmpfuedcu(pfedcu, lstmmpfued)
    lstmmpfnedcu <- listmmpfnedcu(pfedcu, lstmmpfned)
    lstmmpfuedcn <- setdiff(lstmmpfued, lstmmpfuedcu)
    lstmmpfnedcn <- setdiff(lstmmpfned, lstmmpfnedcu)
    fintmrisk<-
      (totalmcrpigmpfu(pmfffued, pmfffudi, pfdicu, pmped,
        pmpfueddicudom, pmpfueddicndom, pmpfunoeddicndom,
        lstmmpfu, lstmmpfuedcu, lstmmpfuedcn)+
      totalmcrpigmpfn(pmfffned, pmfffndi, pfdicu, pmped,
        pmpfneddicudom, pmpfneddicndom, pmpfnnoeddicndom,
        lstmmpfn, lstmmpfnedcu, lstmmpfnedcn))/
    (sum(mfarm)*litter-
      ((totaldeathmmpfu(pmped, lstmmpfuedcu, lstmmpfuedcn)+
        totaldeathmmpfn(pmped, lstmmpfnedcu, lstmmpfnedcn))*
        sum(mfarm)/
        (sum(mfarm[lstmmpfu])+sum(mfarm[lstmmpfn]))
      ))
    return(fintmrisk)
  }

```

中規模農場のモデルを走らせるコード プログレスバーでシミュレーションの進捗状況をモニタリングする。シミュレートされる 5000 回の結果を記憶させ、ファイルに格納する。

```

pbm<-txtProgressBar(min = 0, max = length(nmmp), style = 3)

Bintmrisk <- numeric(length(nmmp))

for (i in seq_along(nmmp)) {

```

```

Bintmrisk[i] <- intmrisk(nmmp[i],pfa[i],
                        pmffued[i],pmffned[i],
                        pmffudi[i],pmffndi[i],
                        pfedcu[i],pfdicu[i],pmped[i],
                        pmpfueddicudom[i],pmpfueddicndom[i],pmpfunoed
dicndom[i],
                        pmpfneddicudom[i],pmpfneddicndom[i],pmpfnnoed
dicndom[i])
  setTxtProgressBar(pbm, i)
}

close(pbs)
save(Bintmrisk, file = "Bintmrisk.Rdata")

```

すると R コンソールにプログレスバーが現れ、以下のように左から右へ延びる。3 秒間ほどでシミュレーション完成。

```

|=====| 7
6% |
|=====| 9
9%
|
|=====| 10
0%

```

シミュレーション結果を表示する。

```

sortmrisk<-sort(Bintmrisk)
mean(sortmrisk)

## [1] 0.05381908

sortmrisk[2500]

## [1] 0.05007061

```



```
sortmrisk[0.25*5000]
```

```
## [1] 0.04050335
```

```
sortmrisk[0.975*5000]
```

```
## [1] 0.1005307
```

```
sortmrisk[5000]
```

```
## [1] 0.1751769
```

3. 大規模農場

```
intlrisk<-function(nlmp,pfa,plffued,plffned,plffudi,plffndi,pfedcu,  
                  pfdicu,plped,  
                  pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
                  pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom){  
  lstlmp <- listlmp(nlmp)  
  lstlmpfu <- listlmpfu(pfa,lstlmp)  
  lstlmpfn <- setdiff(lstlmp,lstlmpfu)  
  lstlmpfued <- listlmpfued(plffued,lstlmpfu)  
  lstlmpfned <- listlmpfned(plffned,lstlmpfn)  
  lstlmpfuedcu <- listlmpfuedcu(pfedcu,lstlmpfued)  
  lstlmpfnedcu <- listlmpfnedcu(pfedcu,lstlmpfned)  
  lstlmpfuedcn <- setdiff(lstlmpfued,lstlmpfuedcu)  
  lstlmpfnedcn <- setdiff(lstlmpfned,lstlmpfnedcu)  
  fintlrisk<-  
    (totalmcrpiglmpfu(plffued,plffudi,pfdicu,plped,  
                      pmpfueddicudom,pmpfueddicndom,pmpfunoeddicndom,  
                      lstlmpfu,lstlmpfuedcu,lstlmpfuedcn)+  
     totalmcrpiglmpfn(plffned,plffndi,pfdicu,plped,  
                      pmpfneddicudom,pmpfneddicndom,pmpfnnoeddicndom,  
                      lstlmpfn,lstlmpfnedcu,lstlmpfnedcn))/  
  (sum(lfarm)*litter-
```

```

    ((totaldeathlmpfu(plped,lstlmpfuedcu,lstlmpfuedcn)+
      totaldeathlmpfn(plped,lstlmpfnedcu,lstlmpfnedcn))*
      sum(lfarm)/
      (sum(lfarm[lstlmpfu])+sum(lfarm[lstlmpfn])))
  ))
  return(fintlrisk)
}

```

大規模農場のモデルを走らせるコード プログレスバーでシミュレーションの進捗状況をモニタリングする。シミュレートされる 5000 回の結果を記憶させ、ファイルに格納する。

```

pbl<-txtProgressBar(min = 0, max = length(nlmp), style = 3)

Bintlrisk <- numeric(length(nlmp))

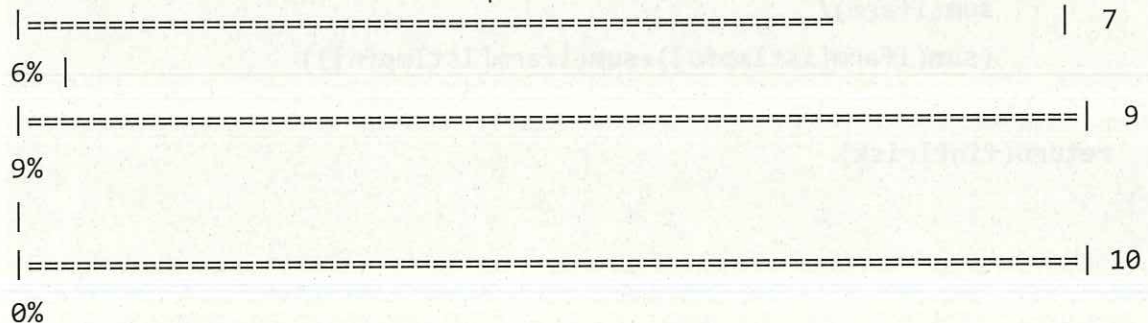
for (i in seq_along(nlmp)) {
  Bintlrisk[i] <- intlrisk(nlmp[i],pfa[i],
    plffued[i],plffned[i],
    plffudi[i],plffndi[i],
    pfedcu[i],pfedcu[i],plped[i],
    pmpfueddicudom[i],pmpfueddicndom[i],
    pmpfunoeddicndom[i],
    pmpfneddicudom[i],pmpfneddicndom[i],
    pmpfnnoeddicndom[i])
  setTxtProgressBar(pbl, i)
}

#close(pbl)

save(Bintlrisk, file = "Bintlrisk.Rdata")

```


すると R コンソールにプログレスバーが現れ、以下のように左から右へ延びる。3 秒間
 ほどでシミュレーション完成。



シミュレーション結果を表示する。

```
sortlrisk<-sort(Bintlrisk)
```

```
mean(sortlrisk)
```

```
## [1] 0.05988623
```

```
sortlrisk[2500]
```

```
## [1] 0.05586294
```

```
sortlrisk[0.25*5000]
```

```
## [1] 0.04540907
```

```
sortlrisk[0.975*5000]
```

```
## [1] 0.1125246
```

```
sortlrisk[5000]
```

```
## [1] 0.1870209
```