Supporting Information:

A dynamic simulation model to support reduction in illegal trade within legal wildlife markets

Appendix S1. Open- ended Key Informant Interviews

Key-informant open-ended semi-structured interviews started with an explanation of the project scope and objectives and followed with a list of reference questions (see below). Interviews were performed during March-April 2019. Due to COVID-19, all interviews were done over the phone or email. Before each interview, we informed participants that participation was voluntary and that participants could refuse to answer any particular question. We used snowballing sampling, starting with leaders of the main fishers' associations in the region and known contacts of the researchers. In total, we interviewed 23 informants, including fishers, government officials, intermediaries, vendors, NGOs staff, and enforcement agents. Depending on the key-informant's role, some questions were omitted, and others were explored in more depth. Interviews over the phone lasted between 30 and 90 minutes. The study complied with Oxford University's ethical requirements (approval number R68516/RE001). These methods represent a subsample of those reported in Oyanedel et al. 2021.

Key-informant interview list of reference questions

a. *Factors affecting actor's decision to trade legal or unreported products*

- What are the main factors that affect the decision to trade legal or unreported/illegal common hake?

- Do intermediaries always carry legal hake? How much?
- What determines the legal/illegal purchase ratio?
- Does this ratio vary?
- What affects its variability?

b. Operation of the legal/unreported market

- Can legal and unreported products be distinguished at the market?
- How does fish enter the market without a permit?
- Is there a price premium for legal products? How much is it?

c. Overarching market dynamic

- What are the most important factors that determine prices?
- Does fishing activity respond to prices? Or, alternatively, are prices driven by the quantities of fish landed

d. Prior ranges

- What is the price of a legal and an illegal unit, at the port and at the market?

Appendix S2. Model runs without and with enforcement data

To test the appropriateness of including weekly enforcement effort data into the model, we first run the model with a randomly generated weekly vector of enforcement effort (with minimum and maximum similar to data). We then run the model again, included the weekly enforcement effort data (average of 2014-2019). We found that including the enforcement data helped better predict the legal landings data dynamics (Figure SM 1). This was especially so for the peaks in legal landings around week 11-16 and 33-37.

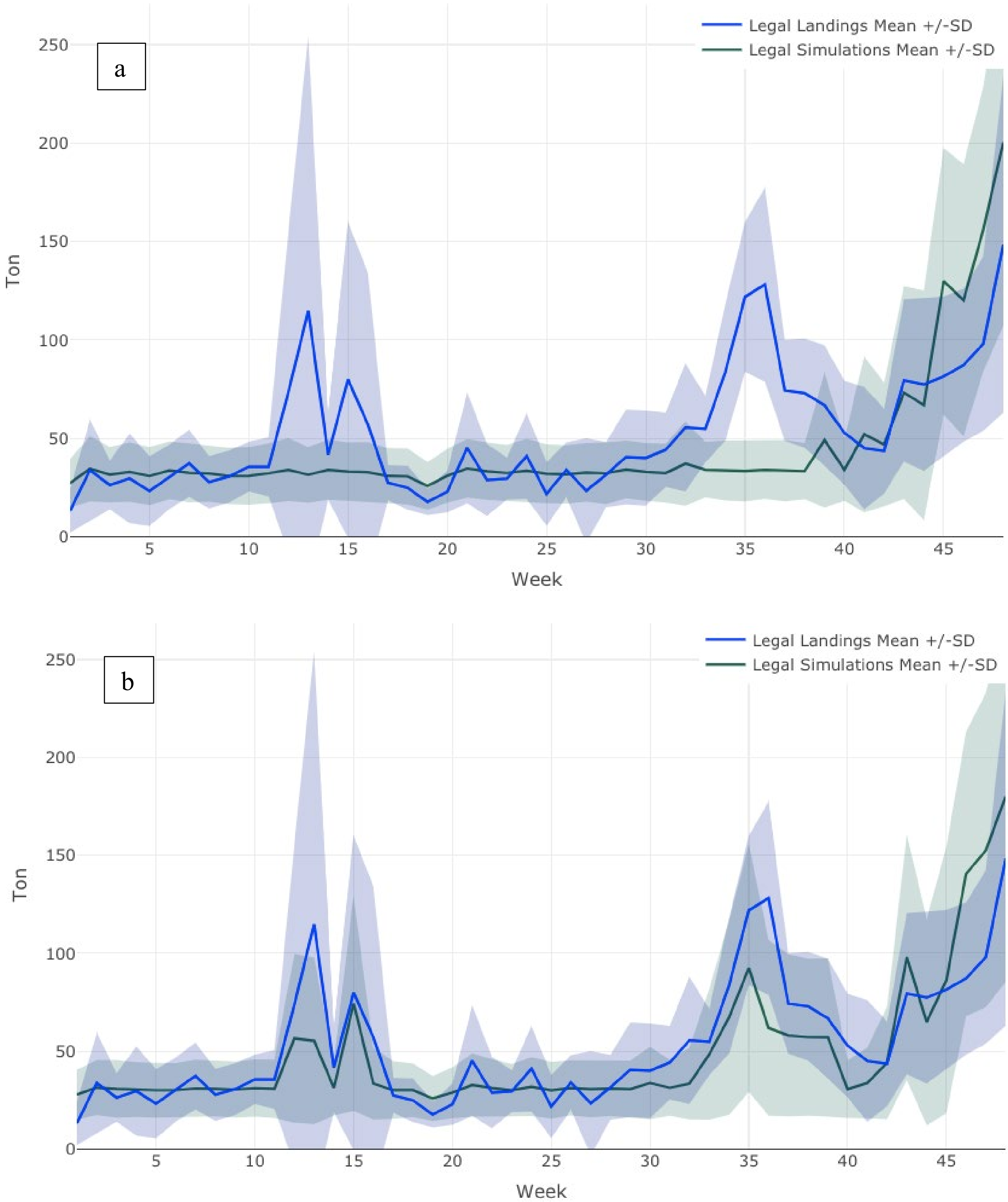


Figure SI 1. Model run without (a) and with (b) weekly enforcement effort data

Appendix S3. Prior range construction and data sources

Table S1. Parameters used in the simulation model case study application and their source.

Parameter	Description	Type	Value	Source
Theta: Efficiency of enforcement (θ_e)	Fraction of illegal units that each enforcement action detects. We fixed this parameter for each simulation, so the same value is used across weeks	Prior – Uniform distribution	Range: 2e-7 - 2e-8 units/enforcement action	Number of activities from enforcement records and overall illegal units traded estimates from Oyanedel et al. 2021. See below for detail
Beta: Price premium (β_l)	Price paid to traders for legal units at the end-market. Unit is Chilean pesos. We fixed this parameter for each simulation	Prior – Uniform distribution	Range: CLP 0-3,000 (4,3 USD)	Range obtained from key informant interviews
Minimum legal fraction per week ($x_{l,m}$)	Percentage set minimum of legal units that traders take each week. We fixed this parameter for each simulation	Prior – Uniform distribution	Range: 10-20% of a truckload	Range obtained from key informant interviews
Price elasticity ($\in_p$)	Elasticity of the price of units at the end-market, depending on units available. We fixed this parameter for each simulation and used the same value for legal and illegal units	Prior – Uniform distribution	Range: proportion from 0 to 1	Broad range used because we obtained no information from key informant interviews
Cost elasticity ($\in_c$)	Elasticity of the cost of units at the port, depending on units available. We fixed this parameter for each simulation and used the same value for legal and illegal units	Prior – Uniform distribution	Range: proportion from 0 to 1	Broad range used because we obtained no information from key informant interviews
Permit fee (V_l)	Value charged to traders for legal units at the ports. Unit is Chilean pesos. We fixed this parameter for each simulation	Data	CLP$ 3,000 (4,3 USD) except towards end of the year	Value obtained from key informant interviews
Permit fee elasticity ($\in_v$)	The rate at which permit fee value decreases towards the end of the year. We fixed this parameter for each simulation	Prior – Uniform distribution with range	Proportion from 0.15 to 0.25	We obtained the range for this parameter through iterations of the model and fit of data, see below for formula
Total units traded (δ)	Sum of legal and illegal units. This parameter changes each week, when we draw random values from the prior	Prior – Uniform distribution	Range: 2000 – 13333 units/week	Upper limit obtained from illegal estimates from Oyanedel et al. 2021. The lower limit is no illegal landings
Legal units traded ($x_{l,d}$)	Data on legal units traded each week	Data – Weekly mean value and SD for 2014-2019 period	See Table S2	From government landings data

Enforcement actions (θ_a)	Data on the number of enforcement actions (officers doing enforcement in the region) each week	Data – Weekly mean value and SD for 2014-2019 period	See Table S2	From government enforcement data
Price reference (P_R)	We used a reference price to calculate price considering elasticity. We used a reference quantity, which we set at the mean of the total landing's prior. We fixed this parameter across weeks and simulations	Data	CLP$ 30,000 (43 USD)	Value obtained from key informant interviews
Cost reference (C_R)	We used a reference cost to calculate price considering elasticity. We also used a reference quantity, which we set at the mean of the total landing's distribution. We fixed this value across weeks and simulations	Data	CLP$ 15,000 (21,5 USD)	Value obtained from key informant interviews
Fixed fine if detected (f_i)	Monetary fixed fine if detected trading illegal units, which we fixed across weeks and simulations	Data	CLP 9,200,00 (13,090 USD)	From government legislation
Fine per illegal unit constant (c_i)	Monetary fine if detected trading illegal units, per unit, fixed across weeks and simulations	Data	$4P_l$	From government legislation
Overall and stage-specific quota (Q)	Total quota for legal trading in a year, which we fixed across simulations. The government gives this quota in 3 periods within the year	Data – Mean value for 2014-2019	98,000 units. See below for periods	From government registers

a. *Efficiency of enforcement (θ_e)*

For the efficiency of enforcement (θ_e) parameter, the lower range of the prior was calculated as:

$$\theta_e = x_c/(\theta_E n_T i_r)$$

Where: x_c= total amount of units confiscated by the enforcement authority during the 2014-2019 period (10,230); θ_E= sum of the effort in terms of number of enforcement visits during the 2014-2019 period (1,538); n_T= total units traded (3,788,373) considering legal catch, and using i_r from Oyanedel et al. 2021 that illegal rate maximum is 88%. For the upper range of the prior, we multiplied this lower range value by ten, to account for a broad dispersion of how much the real value of θ_e could be.

b. Permit fee (V_l) equation

Based on the permit fee elasticity ($\in_v$), we calculated V_l in the model as follow:

$$V_l = max(0, V_r(1 - \in_v(max\ (0, R - R_0)\ /R_0)))$$

Where: V_r= permit fee reference value (CLP 3,000); $R = ql_t/(T\text{-t+1})$; $R_0 = Q/(\text{T+1})$; $ql_t =$ quota left in time t; $Q =$ overall quota; T= time horizon.

c. Overall and staged quota (Q)

The overall quota for the 2014-2019 period mean was 98,000 units, given in stages, according to government data as:

- January= 8,613
- February to June=49,000
- July to December=98,000

d. Data sets used

Table S2. Data sets used: landings mean (+/- SD) per week and enforcement (mean) count of visits per week in the region.

Week	Landings			Enforcement
	Mean	Mean + SD	Mean - SD	Mean
1	493	902.49	83.51	1.33
2	1258	2203.79	312.21	4.00
3	977	1434.60	519.40	6.00
4	1101	1942.55	259.45	4.50

5	862	1518.96	205.04	2.33
6	1134	1739.31	528.69	4.17
7	1388	2019.79	756.21	5.00
8	1031	1521.88	540.12	5.50
9	1137	1627.28	646.72	4.83
10	1323	1791.84	854.16	4.83
11	1320	1874.06	765.94	5.67
12	2731	5767.54	-305.54	9.67
13	4254	9402.74	-894.74	9.33
14	1544	2389.20	698.80	4.17
15	2964	5935.31	-7.31	11.83
16	2117	4973.98	-739.98	6.17
17	1022	1353.31	690.69	6.33
18	927	1340.03	513.97	3.17
19	657	900.70	413.30	3.17
20	851	1241.07	460.93	4.33
21	1675	2715.87	634.13	6.00
22	1072	1749.47	394.53	5.67
23	1098	1496.17	699.83	4.33
24	1522	2329.96	714.04	5.67
25	812	1409.37	214.63	2.67
26	1264	1780.41	747.59	4.33
27	870	1859.05	-119.05	2.17
28	1173	1785.84	560.16	3.33
29	1504	2398.33	609.67	3.83
30	1486	2384.21	587.79	5.33
31	1639	2336.42	941.58	3.67
32	2060	3264.83	855.17	4.83
33	2030	2656.30	1403.70	7.17
34	3099	4386.78	1811.22	8.50
35	4517	5926.72	3107.28	10.83
36	2091	4705.74	-523.74	7.50
37	4314	6070.10	2557.90	7.50
38	2647	3659.65	1634.35	7.33
39	2590	3331.04	1848.96	7.17
40	2316	3352.96	1279.04	3.00
41	1760	2829.25	690.75	4.00
42	1549	2439.28	658.72	4.83
43	2336	4086.18	585.82	7.33
44	3071	4191.20	1950.80	4.33
45	2896	4470.63	1321.37	3.50

46	2868	4262.47	1473.53	5.83
47	3532	5136.05	1927.95	5.50
48	5350	8660.61	2039.39	3.50

Appendix S4. Mahalanobis distance distribution

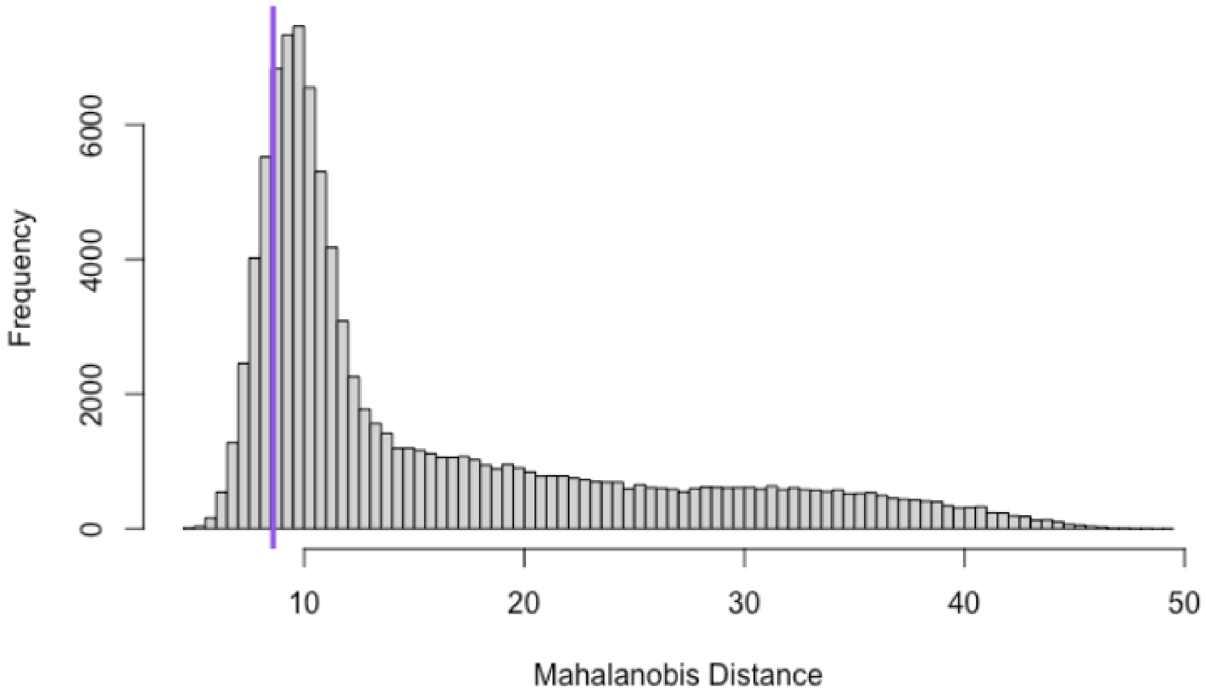
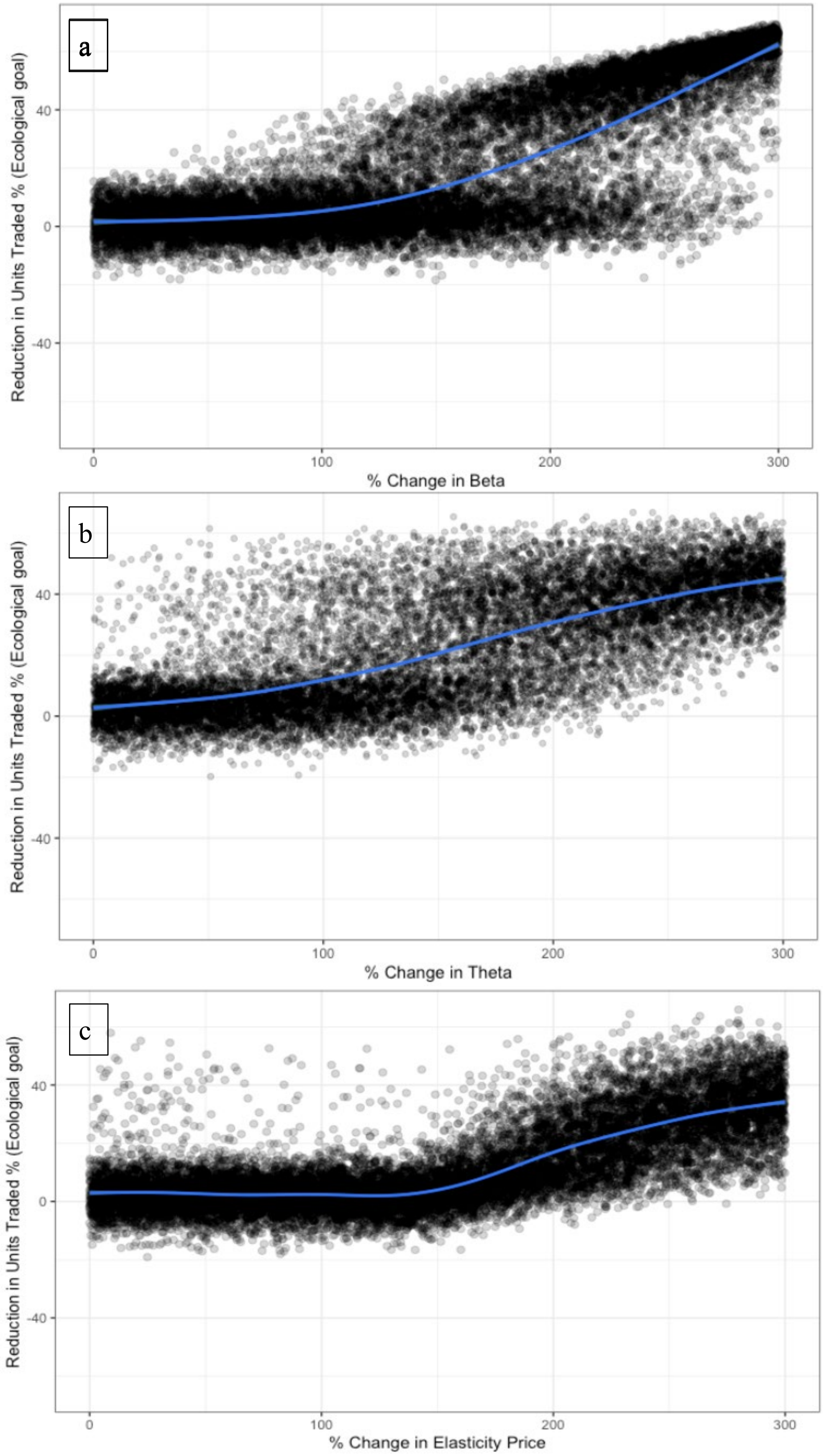


Figure SI 2. Mahalanobis distance distribution of model results. Purple line indicates threshold, so simulations to the left are accepted (~15%).

Appendix S5. Sensitivity Analysis



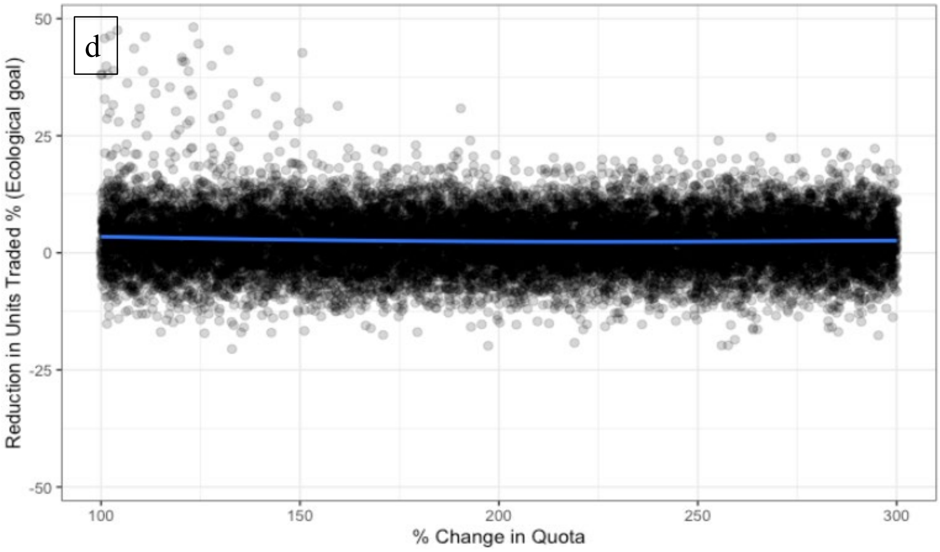


Figure S3. Evaluation of ecological sustainability of the fishery improvements (as measured by decreases in units traded) when changing the value of parameters that are proxies for possible policy levers: a) theta (efficiency of enforcement); b) beta (price premium); c) price elasticity, and d) quota. Black dots represent simulation results, and the blue line is the smoothed conditional mean.

R Code

A dynamic simulation model to support reduction in illegal trade within legal wildlife markets

Load required packages

````
```{r, cache=FALSE}

library(ggplot2)
library(plotly)

```
````

Load data

````
```{r, cache=FALSE}

legal_units_data   <-  #Here load a vector of data for legal units traded
(mean). For instance, use per week value. If no data available can use the
following to try the model read.csv('https://raw.githubusercontent.com/XXX
/illegal_fishing/master/data/unitsYear.csv')
enforcement_data <-   #Same than units, could use this dataset for demonst
ration: read.csv ('https://raw.githubusercontent.com/XXX/illegal_fishing/m
aster/data/enfcomnocom.csv')

```
````

Setting variable

````
```{r, cache=FALSE}

###Setting variables for the model. All monetary values are in Chilean pes
os
iterations= 10000    ###  Number of times the model is ran
weeks= 48            ###  Weeks in a normal year, for demonstration purpos
es we use 48 which excludes September when there is no fishing in our case
study
legal_units_data  = legal_units_data[1:weeks,]
cost = 500           ###  Approximate per unit traded
fine = 9.2e+05       ###  Fine expected if detected trading illegal produc
ts
permit_fee = 3000    ###  Use if there is a cost difference between legal
and illegal products at the harvest level
c_ref= 15000         ###  Reference cost per unit
p_ref= 30000         ###  Reference price per unit
mean_land=mean(legal_units_data$mean)*weeks

#### Creates a vector of quota available in each week. Can be the same val
````
````

````
ue across weeks, or can be given in stages (see Supplementary Material)
quota_overall= 98000 ###  Yearly overall quota for the species
quota_available          = (1:weeks)
quota_available[1:weeks] = quota_overall

#### Create a data frame of enforcement activities mean per week
enforcement=as.data.frame((enforcement_data[,3]))
```
````

Create priors for Approximate Bayesian Approach

````
```{r, cache=FALSE}
###First, create ranges
####Prior range for beta (price premium)
beta_min=0    ### price premium lower limit
beta_max=3000 ###price premium higher limit

####Prior range for overall units trade
###First define overall range of units traded and how much are illegal uni
ts traded ratio
MaxIllegalrate=0.85    #85% Illegal, higher limit (this can be changed if
there is an estimate of % of illegal units traded in the market)
MinIllegalrate=0.0     #0% Illegal, lower limit

###Then calculate Prior range
T_min=(quota_overall/weeks) * (1/(1-MinIllegalrate))  ## Min overall units
traded
T_max=(quota_overall/weeks) * (1/(1-MaxIllegalrate))  ## Max overall units
traded
qr=(T_min+T_max)/2                                    ## Reference quantit
y for elasticities

####Prior range for Theta (probability of detection per unit for each enfo
rcement action)
theta_max= 2e-7    ### See supporting information
theta_min= 2e-8

###Prior range for cost elasticity
E_minC=0
E_maxC=1

###Prior range for price elasticity
E_minP=0
E_maxP=1

###Prior range for end of year permit_fee elasticity
permit_feeend_min=0.15   ### See supporting information, but this might no
t be necessary outside of the case study. If there is no need for this par
ameter, then use 0
permit_feeend_max=0.25

````

````
###rior range for minimum legal per week
minlegal_min=0.1   ### See supporting information
minlegal_max=0.2

####This creates the priors based on the ranges above
beta_P                 = runif(n=iterations, min = beta_min, max = beta_max)
theta_P                = runif(n=iterations, min=theta_min,  max=theta_max)
T_I                    = runif(n=iterations, min=T_min,  max=T_max)
permit_fee_elasticity_P  = runif(n=iterations, min=permit_feeend_min,  max
=permit_feeend_max)
Price_elasticity_P = runif(n=iterations, min=E_minP,  max=E_maxP)
Cost_elasticity_P  = runif(n=iterations, min=E_minC,  max=E_maxC)
min_legal_P            = runif(n=iterations, min=minlegal_min,  max=minlegal_m
ax)
```
````

Run Simulation model

````
```{r, cache=FALSE}
###Tracks time it takes to run
start_time <- Sys.time()
start_time

##Create matrix for results
results          = matrix(0,iterations,14)     ### Keeps track of simulati
on results
legal_units      = matrix(0,weeks,iterations)  ### Keeps track of legal un
its in each simulation
illegal_units    = matrix(0,weeks,iterations)  ### Keeps track of illegal
units in each simulation
permit_fee_value = matrix(0,weeks,iterations)  ### Keeps track of permit_f
ee value in each simulation
Quota_left       = matrix(0,weeks,iterations)  ### Keeps track of quota le
ft in each simulation
ratio            = matrix(0,weeks,iterations)  ### Keeps track of quota le
ft in each simulation

### Start loop of iterations
for (simulation in 1:iterations) ### Start iterations
{
  ###Sample each parameter from the prior

  beta                      = sample(beta_P,1)
  theta                     = sample(theta_P,1)
  price_elasticity          = -(sample(Price_elasticity_P,1))
  cost_elasticity           = -(sample(Cost_elasticity_P,1))
  permit_fee_elasticity     = (sample(permit_fee_elasticity_P,1))
  min_legal                 = (sample(min_legal_P,1))

  ##Start loop of weekly simulations
  weekly_simulation   =  matrix(0,weeks,11)
````

```
  for (t in 1:weeks)
  {
  prob_detection = theta*enforcement[t,1] # uses the enforcement data per week, otherwise enforcement could be a vector of 1s of length weeks
  x_T             = sample(T_I,1)          # Sample units for that week

  ###Calculates quota left for end of year and value of permit_fee in t
  quota        = quota_available[t]
  quota_left   = (quota-sum(weekly_simulation[,1]))
  rate_r       = quota_overall/(weeks+1)
  rate         = quota_left/(weeks-t+1)
  permit_fee_t= max(0, permit_fee * (1 - permit_fee_elasticity * (max(0, rate - rate_r) / rate_r)))

  #Calculates prices and costs
  C_L=   (c_ref *  (1-((cost_elasticity*((qr-x_T)/qr)))))  + permit_fee_t #Elasticity of cost for legal boxes, wrt units + permit_fee
  C_I=   (c_ref *  (1-((cost_elasticity*((qr-x_T)/qr))))) #Elasticity of cost  for illegal boxes, wrt units
  P_L=   (p_ref  * (1-((price_elasticity*((qr-x_T)/qr))))) + beta #Elasticity of price legal boxes, wrt units + beta
  P_I=   (p_ref  * (1-((price_elasticity*((qr-x_T)/qr))))) #Elasticity of price illegal boxes, wrt units

  ###Calculates x_l (legal units), x_i (illegal units) with constraint that x_l cannot be higher than x_T, or lower than 0 and there is no trading if there is no quota left
  x_l          = x_T-((((P_I-C_I-P_L+C_L-(fine*  prob_detection))/(8*  prob_detection*(P_L))))) ##Legal units (here we use 8 insted of 2 as indicated in the manuscript, because fines in Chile the variable fine is 4P_L (Supplementary Material)
  min_bound   = if(quota_left>0) {min_legal * x_T} else 0
  max_bound   = min(x_T, quota_left)
  x_l         = if(x_l <= max_bound) {x_l} else {max_bound}
  x_l         = if(x_l <= min_bound) {min_bound} else {x_l}
  x_i         = if(quota_left>0) {x_T-x_l} else 0


  weekly_simulation[t,1] = x_l
  weekly_simulation[t,2] = x_i
  weekly_simulation[t,3] = C_L
  weekly_simulation[t,4] = C_I
  weekly_simulation[t,5] = P_L
  weekly_simulation[t,6] = P_I
  weekly_simulation[t,8] = permit_fee_t
  weekly_simulation[t,9] = quota_left
  weekly_simulation[48,9] = quota_left - x_l
  weekly_simulation[t,10] = (x_i/(x_i+x_l))
  weekly_simulation[t,11] = (x_l*C_L) + (x_i*C_I)

  }
```

```

  ### Fill matrices for posterior analysis
  legal_units[,simulation]       = weekly_simulation[,1]
  illegal_units[,simulation]     = weekly_simulation[,2]
  permit_fee_value[,simulation] = weekly_simulation[,8]
  Quota_left[,simulation]        = weekly_simulation[,9]
  ratio[,simulation]             = weekly_simulation[,10]

  ## Fill results of each simulation, by summing across weeks and capture
prior values
  results[simulation,1] = sum(weekly_simulation[,1])
  results[simulation,2] = sum(weekly_simulation[,2])
  results[simulation,3] = sum(weekly_simulation[,1])+sum(weekly_simulation
[,2])
  results[simulation,4] = beta
  results[simulation,5] = theta
  results[simulation,6] = price_elasticity
  results[simulation,7] = cost_elasticity
  results[simulation,8] = permit_fee_elasticity
  results[simulation,9] = sum(weekly_simulation[,2])/(sum(weekly_simulatio
n[,1])+sum(weekly_simulation[,2]))  ## Ratio
  results[simulation,10]= simulation
  results[simulation,12]= min_legal

}


end_time <- Sys.time()
end_time - start_time
```

Mahalanobis distance rejection criteria calculation

```
```{r, cache=FALSE}

#Create matrices and units values needed to calculate Mahalanobis distance
SimResults          = data.frame(results)
mahalanobis_dist    = matrix(0,iterations,2)
true_mean           = apply(legal_units_data, 1, mean, na.rm=TRUE)
true_std            = apply(legal_units_data, 1, sd, na.rm=TRUE)
true_cov            = diag(true_std ** 2)


#Calculate the Mahalanobis distance for each simulation result
for (m in 1:iterations)
{
mahalanobis_dist[m,1]= sqrt(mahalanobis(legal_units[,m], true_mean, true_c
ov))
mahalanobis_dist[m,2]=m
}

##Reject those simulations above the threshold
```
```

```
p=0.95                    ## Rejection criteria
threshold                 = sqrt(qchisq(p=p, df=weeks))
mahalanobis_dist_filter= subset(mahalanobis_dist,mahalanobis_dist[,1]<thre
shold)

## Plot distances and threshold (line)
hist       (mahalanobis_dist[,1], breaks=100, xlab="Mahalanobis Distance",
ylab="Frequency of Simulations")
abline     (v=threshold, col="purple",  lwd=3)


## Select those simulations that were accepted
Accepted_Legs        = legal_units[,c(mahalanobis_dist_filter[,2])]      #
#Select those legal vectors that passed the filter
Accepted_Ills        = illegal_units[,c(mahalanobis_dist_filter[,2])]    #
#Select those illegal vectors that passed the filter

filter          = mahalanobis_dist_filter[,2]
SimResults      = subset(SimResults,SimResults$X10 %in% filter)    ##Select
parameter values of those simulations that passed the filter

#Create posteriors variables for graphs
Legalboxes      = SimResults$X1
Illegalboxes    = SimResults$X2
unitsboxes      = SimResults$X3
Beta            = SimResults$X4
Theta           = SimResults$X5
ElasticityPrice= SimResults$X6
ElasticityCost = SimResults$X7
permit_feeEndofyear  = SimResults$X8
Minlegal        = SimResults$X12
Ratio           = SimResults$X9
```

Parameter Posterior Graphs

```{r, cache=FALSE}
###This code creates the graphs for the posterior distributions, figure 3

####Theta
options(scipen=999)
old.par <- par(mfrow=c(2, 3))
DDetIn    = density(Theta, n=iterations, adjust=3,from=theta_min, to=theta_
max) #
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Theta (efficiencty of enforcement)"
,ylab="Probability density", col="black", bty="n")
```

```r
###Beta
DDetIn   = density(Beta, n=iterations, adjust=3, from=beta_min, to=beta_max)#
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Beta (price premium)",ylab="Probability density", col="black", bty="n")


###Min Legal
DDetIn   = density(Minlegal, n=iterations, adjust=3, from=minlegal_min, to=minlegal_max)#
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Minimum legal fraction per week",ylab="Probability density", col="black", bty="n")



###Price Elasticity
DDetIn   = density(-ElasticityPrice, n=iterations, adjust=3, from=0, to=1) #from=E_minP, to=E_maxP)
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Price elasticity",ylab="Probability density", col="black", bty="n")



###Cost Elasticity
DDetIn   = density(-ElasticityCost,   n=iterations, adjust=3, from=0, to=1) #from=E_minC, to=E_maxC
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Cost elasticity",ylab="Probability density", col="black", bty="n")


##permit_fee Elasticity
DDetIn   = density(permit_feeEndofyear, n=iterations, adjust=3,  from=permit_feeend_min, to=permit_feeend_max)
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Permit fee elasticity",ylab="Probability density", col="black", bty="n")

##Total units
old.par <- par(mfrow=c(2, 2))
DDetIn   = density(unitsboxes, n=iterations, adjust=3)
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
plot(xDetecIN,yDetecIN, type="l",xlab="Total units traded",ylab="Probability density", col="black", bty="n")

```

````

##Ratio
DDetIn    = density(Ratio, n=iterations, adjust=3)
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
p7        = plot(xDetecIN,yDetecIN, type="l",xlab="Ratio of illegal to tota
l units",ylab="Probability density", col="black", bty="n")

##Illegalboxes
DDetIn=density(Illegalboxes, n=iterations, adjust=4)
SimY=DDetIn$y
SimX=DDetIn$x
plot(SimX,SimY, type="l",xlab="Illegal units",ylab="Probability density",
col="black", bty="n")


## Legal Boxes
DDetIn    = density(Legalboxes, n=iterations, adjust=4)
yDetecIN = DDetIn$y
xDetecIN = DDetIn$x
p8        = plot(xDetecIN,yDetecIN, type="l",xlab="Legal units",ylab="Proba
bility density", col="black", bty="n")
```
````

Graph units over time and comparison to data

````
```{r, cache=FALSE}
#This code creates figure 4

##Calculate weekly means
unitsMean     = transform(Accepted_Legs, MEAN=apply(Accepted_Legs,1, mean,
na.rm = TRUE))
IllegalMean   = transform(Accepted_Ills, MEAN=apply(Accepted_Ills,1, mean,
na.rm = TRUE))
unitsSD       = transform(Accepted_Legs, SD  =apply(Accepted_Legs,1, sd, na
.rm = TRUE))
IllegalSD     = transform(Accepted_Ills, SD  =apply(Accepted_Ills,1, sd, na
.rm = TRUE))

###Create data frame with units data, mean and SD
units_plot          = matrix(0,weeks,9)
units_plot[,1]      = legal_units_data$mean
units_plot[,2]      = legal_units_data$mean_p_std
units_plot[,3]      = legal_units_data$mean_m_std

###Calculate means and SD for simulations legal
units_plot[,4]      = unitsMean$MEAN
units_plot[,5]      = unitsMean$MEAN+unitsSD$SD
units_plot[,6]      = unitsMean$MEAN-unitsSD$SD

###Calculate means and SD for simulations illegal
````

```
units_plot[,7]       = IllegalMean$MEAN
units_plot[,8]       = IllegalMean$MEAN+IllegalSD$SD
units_plot[,9]       = IllegalMean$MEAN-IllegalSD$SD
units_plot           = as.data.frame(units_plot)
boxtoton             = 27/1000

###Creates Graph
##Legal units simulations
units_Figure <- plot_ly(units_plot, x = ~seq(1:weeks), y = ~units_plot$V4*
boxtoton, type = 'scatter', mode = 'lines',
               line = list(color='rgb(0,100,80)'),
               name = 'Legal Simulations Mean +/-SD')
units_Figure <- units_Figure %>% add_trace(y = ~units_plot$V5*boxtoton, ty
pe = 'scatter', mode = 'lines',
                         line = list(color = 'transparent'), name = 'High
units',showlegend = FALSE)

units_Figure <- units_Figure %>% add_trace(y = ~units_plot$V6*boxtoton, ty
pe = 'scatter', mode = 'lines',
                         fill = 'tonexty', fillcolor='rgba(0,100,80,0.2)',
line = list(color = 'transparent'),
                         showlegend = FALSE, name = 'Low units')

##Data
units_Figure <- units_Figure%>% add_trace(units_plot, x = ~seq(1:weeks), y
= ~units_plot$V2*boxtoton, type = 'scatter', mode = 'lines',
                      line = list(color = 'transparent'),
                        showlegend = FALSE, name = 'High units')
units_Figure <- units_Figure %>% add_trace(y = ~units_plot$V3*boxtoton, ty
pe = 'scatter', mode = 'lines',
                        fill = 'tonexty', fillcolor='rgba(0,17,157,0.2)',
line = list(color = 'transparent'),
                        showlegend = FALSE, name = 'Low units')
units_Figure <- units_Figure %>% add_trace(y = ~units_plot$V1*boxtoton, ty
pe = 'scatter', mode = 'lines',
                        line = list(color='blue'),
                        name = 'Legal units Mean +/-SD')


##Illegal units simulations
units_Figure <- units_Figure%>% add_trace(units_plot, x = ~seq(1:weeks), y
= ~units_plot$V8*boxtoton, type = 'scatter', mode = 'lines',line = list(co
lor = 'transparent'),showlegend = FALSE, name = 'High units')
units_Figure <- units_Figure %>% add_trace(y = ~units_plot$V9*boxtoton, ty
pe = 'scatter', mode = 'lines',fill = 'tonexty', fillcolor='rgba(220,20,60
,0.2)', line = list(color = 'transparent'),showlegend = FALSE, name = 'Low
units')
units_Figure <- units_Figure %>% add_trace(y = ~units_plot$V7*boxtoton, ty
pe = 'scatter', mode = 'lines',line = list(color='red'),name = 'Illegal Si
mulation Mean +/-SD')


units_Figure <- units_Figure %>% layout(yaxis = list(range = c(0,270)))
units_Figure <- units_Figure %>% layout(showlegend = TRUE)
```

````
units_Figure <- units_Figure %>% layout(xaxis = list(title = "Week of the
year"))
units_Figure <- units_Figure %>% layout(legend = list(x =0.67, y = 1))
units_Figure <- units_Figure %>% layout(yaxis = list(title = "Quantity tra
ded (tons)"))
units_Figure

```
````
````