data_extracting

Marina Chaji

3/7/2022

Project setup

Load packages. Install if necessary.

```
# Set Path
here::i_am("data_extracting.rmd")
```

```
## here() starts at /net/home2/mlee/Effort-Displacement---Scallop
```

```
PKG <- c("RODBC", "RODM","here","readxl")
for (p in PKG) {
  if(!require(p,character.only = TRUE)) {
    install.packages(p)
    require(p,character.only = TRUE)}
}
```

```
## Loading required package: RODBC

## Loading required package: RODM

## Loading required package: here

## Loading required package: readxl
```

```
vintage_string<-Sys.Date()
vintage_string<-gsub("-","_",vintage_string)
```

Organization

We will:

1. Try to avoid copying data; when we rely on data from other people, we will read it directly into memory from the network location or Oracle.
2. Sometimes this is unnecessary, so we will copy external data into the "data/external" folder. We will have a separate subfolder for shapefiles.
3. Store an intermediate data product in "data/intermediate".
4. Store final data products in "data/main."
5. Use a vintage "suffix" to denote when we have extracted data.

Read in oracle passwords and set network directory

This is a block of code where we set up the oracle passwords and make R aware of folders on the network.

```r
source(here("credentials.R"))

# Set the network_location_desktop and network_location_remote variables somewhere OUTSIDE of this code

#Comment one of these out, depending on whether you are running this code on a server or locally (with
net<-network_location_desktop
net<-network_location_remote

# These are not part of the project path
offshoreWind_directory<-file.path(net,"home5", "dcorvi","OffshoreWind","offshoreWind4","data")
spacepanels_directory<-file.path(net,"home2", "mlee","dropoff","wind")
cost_directory<-file.path(net,"work5","socialsci","Trip_Costs","2007-2020")

# Set up paths.
East_Cst_crop_2020_path<- here("data","external","shapefiles","East_Cst_crop_2020_extended")
TMSQ_path<-here("data","external","shapefiles","Ten Minute Squares Cut North and Greater Atlantic")
All_Lease_Areas_Shapefile_path<-here("data","external","shapefiles","All_Lease_Areas_Shapefile")
```

Purpose

This code extracts and processes data. Our goal is to construct a dataset that can be used to estimate a location choice model at the trip level for the Limited Access Scallop Fishery, using data from 2007-2019 (calendar years). The main datasource is a frozen DMIS table.

Dependencies

This code depends on:

1. Network access to get the APSD_DMIS_2.rda and trip cost data from places on the NEFSC network.
2. The ability to connect to NEFSC oracle databases (VTR and the Live DMIS tables at APSD.t_ssb_trip_current@garfo_nefsc)

Data Overview

There are four main data sources (so far). None are perfect.

Source	Name
DMIS_APSD_2	DMIS is a Northeast Regional Office data record matching system. Primary data sources include Allocation Management System (AMS) Database, Vessel Trip Reports (VTRs), Dealer Reports, Vessel Monitoring System (VMS) Catch Reports, Observer Reports, Vessel Permit Database, and the MQRS database, which tracks limited access fishing eligibilities
VTR (Vessel Trip Reports)	A vessel trip report (VTR) must be received by NMFS or postmarked within 15 days after the reporting month's end. For vessels that also hold a NE multispecies permit, VTRs must be submitted weekly by Tuesday of the week after the fishing trip ends. Copies of VTRs must be retained on board the vessel for 1 year after the last entry on the log and otherwise retained for 3 years after the date of the last entry on the log. If no fishing activity occurred during a reporting period (week or month), then a VTR must be submitted stating that no fishing trips were taken.

Source	Name
Vessel Monitoring System (VMS)	All vessels issued a Federal scallop permit are required to have an active VMS unit and must use their VMS unit to declare all vessel activity, including fishing trips and transiting.
Cost data	Werner et al predict estimate a model of trip costs. Predictions (in and out of sample) are used

We have decided to use the DMIS as our primary dataset. DMIS primarily uses Vessel Trip Reports (VTRs) for "trip" and "effort" data and dealer databases for landings. A drawback of using these data are that there is a single point (latitude and longitude) for each time a vessel deploys a particular type of gear into a statistical area. In the LADAS scallop fleet, vessels rarely, if ever, will switch gears at sea. So, a trip is most likely to have multiple VTRs if it switches statistical areas.

There aren't any big incentives (yet) to misreport statistical areas in Scallop. Unlike groundfish, scallop open areas are all managed with one control (Days-at-Sea). And fishing in the Access Areas, but reporting open areas could occur. But vessels need to declare in, so this is a very risky proposition if you are caught fishing in an Access Area, but declared into an open area.

By choosing to represent the trip as a single point, or as inside a homogeneous ten minute square, we may not have the ability to answer our research question. Alternatively, do we have the ability to model at the sub-trip level?

Other possibilities were considered for our primary dataset:

1. Observer cover a subset of the fishery. According to the 2021 SBRM report, it was approximately 8-10% of effort for the Limited access fleet. This would provide haul level lat-lon and estimates of catch for the sampled subset. We viewed the subset as too limited - it would provide us with observations of approximately 200 Access area and 100 open area trips per year.Observer data contains the sailing and landing port.
2. VMS - VMS data would provide lat-lon at a high frequency. Other researchers have used this; however we uncomfortable with figuring out how to allocate catch along the VMS track. VMS data contains the sailing and landing port.
3. Rasters. The raster data are an intermediate data product that combines trip report with a statistical model describes the distance between observed hauls and the vtr point location. This allows for a smoothing of effort catch across a non-arbitrary grid (like a 10 minute square, statistical area, or just a lat-lon point).

DMIS

We are using the DMIS_APSD_2 table.

Column	Description
DOCID	VTR DOCUMENT table record identifier; Primary key, internally generated at scanning based on vessel id and date/time sailed. Each DOCID represents one trip; equivalent to TRIPID in VESLOG tables.
Dates	VTR land date, AMS Land Date, Dealer Sold Date Trip date is broken down into fields Calendar_Year, Month_of_Year, Week_of_Year, and Day_of_Year
DDLAT	Latitude in decimal degrees
DDLON	Longitude in decimal degrees
PERMIT	Six-digit vessel fishing permit number assigned by the NE Regional Office permit system

Column	Description
DOLLAR	This is the value of fish sold.An imputed price is used in cases where the value was not reported.
POUNDS	POUNDS is live weight, (in the shell)
LANDED	LANDED can be meat weights or shell weights, but is usually meats
TRIP_LENGTH	Trip length is in days; It is calculated from the elapsed time between the date-time sailed and date-time landed. This is a measure of days absent.

Ports – currently assuming that a trip departs from the same place it lands. An alternate assumption is that people just leave from the place where they made their last landing.

The DDLAT and DDLOn are self reported lat-lons from logbooks(VTRs). We have supplemented this with some extra information.

Column	Description
TRIP_ID	
DOCID	VTR DOCUMENT table record identifier; Primary key, internally generated at scanning based on vessel id and date/time sailed. Each DOCID represents one trip; equivalent to TRIPID in VESLOG tables.
ACTIVITY_CODE	Complicated set of letters and numbers. See below
PLAN_CAT	LGC_A LGC_B, LGC_C, SC_2, SC_3, SC_4, SC_5, SC_6, SC_7, SC_8, SC_9, SG_1A, SG_1B are a collection of true and false variables the indicate if the vessel had a particular permit when the trip was taken.

ACTIVITY_CODE and a set of PLAN_CAT categorical variables are used from the live DMIS tables. We join using TRIP_ID, DOCID. See here

Description of the VTR data.

Column	Description
TRIPID	VESLOG Trip record identifier, which is generated internally and used for linking
tripcatg	(only commercial categories are selected), recreational and RSA/EFP are not.
operator	Name of the captain
opernum	Captains Identification number
permit	Six-digit vessel fishing permit number assigned by the NE Regional Office permit system
nsubtrip	Number of subtrips (see description of subtrips
crew	number of crew, including captain
port	6 digit numeric code for the port, renamed to VTR_PORTNUM to make clear is a companion to VTR_PORT and VTR_STATE

Description of the SPACEPANELS data.

Column	Description
TRIPID	VESLOG Trip record identifier, which is generated internally and used for linking
geoid	10 digit county subdivision from US Census.
state_fips	2 digit state fips code

Column	Description
portlnd1	string of the name of the port that the vessel operator writes on the VTR.
state1	2 letter abbreviation of the state
port	6 digit port code. Should match vtr_portnum from VTR
namelsad	Name and Legal/Statistical description
port_lat	Latitude of the geoid
port_lon	longitude of the geoid
previous_geoid	Geoid of landing port for previous trip
previous_state_fips	2 digit state fips code for previous trip
previous_namelsad	Name and Legal/Statistical description for previous trip
previous_port_lat	Latitude of the geoid for previous trip
previous_port_lon	longitude of the geoid for previous trip

The spacepanels data tidies up vtr ports and aggregates them to the US Census county subdivision. You should *not* expect an exact match between portlnd1, state1, and port in the spacepanels dataset compared to the same columns in the raw vtr because some data clean was done on these fields. The code is in the spacepanels repo, "just_ports.do."

- `Geoid` is geoid10. The lat-lons are either the centroid of the geoid and/or adjusted to the coast. There is probably some error here, but if the goal is to use these points to help construct distances or costs to go fishing, they are probably accurate enough.
- `namelsad` is a convenient name (Like "Boston city"). However, be aware that there are some places with the same name, so use either the `geoid` or the `namelsad` plus `state` or `state_fips`. This could also be solved by using the `namelsad` as factor levels This dataset includes all trips from 1996-2021. Use the vintage date appended to the end of the file to assess whether the final year of data is "complete" enough for your purposes. The following corresponds to all trips, not just scallop trips:
- Missing `tripids`. there are missing tripids prior to 2003. These correspond to SCOQ.

- Missing `vtr_portnum` There are about 183 obs. These are 2019-2021 and probably reflect changes in the underlying data that haven't been picked up and cleaned in the code. I am not going to deal with these.
- Missing `geoid`. There are about 3,000 obs. Many of these are because the vtr_port is "Other State". A few are missing because of a new port.
- There is a set of `previous_` variables. These were constructed using this code:

```
bysort permit (datelnd1 tripid): gen previous_geoid=geoid[_n-1]
bysort permit (datelnd1 tripid): gen previous_namelsad=namelsad[_n-1]
bysort permit (datelnd1 tripid): gen previous_state_fips=state_fips[_n-1]
```

- It's possible that the previous trip was the same day. It's also possible that the previous trip was from years before.
  *The date that I am using here is the `datelnd1` field from VESLOG_T, if those are somehow missing, I have used datesold from VESLOG_S. I am using something from the Clam logbooks, but I'm not positive as to what. I normally used the datesold in VESLOG_S for spacepanels.
- More of the `previous_` variables are missing. This is expected, because the first observation of a permit will be missing something. I don't think this will cause too much of a problem. Some will be missing if the prior trip was a "other state."

Description of the VMS data.

We are currently not using the VMS data directly.

Description of the Cost data.

Column	Description
Place	holder
Place	holder

Load Offshore Wind Tool Data sets

The frozen DMIS table from the offshoreWind project (APSD_DMIS_2) is the base dataset for the analysis. The DMIS data are formed by combining many datasets, including VTR and Dealer. In brief, the APSD_DMIS_2 dataset contains a mix of trip attributes (port, date), sub-trip attributes (gear, location) , and catch outcomes (species, pounds, landed, dollar). You can read more about DMIS here.

```
load(file.path(offshoreWind_directory, "APSD_DMIS_2.rda"))
```

Read in Port lats and lons

Load in the lats and lons of all ports from the spacepanels directory. Change the column names to clarify that the lat-lons are the ports. Join this to the APSD_DMIS_2 dataset, keeping all rows of the APSD_DMIS_2 dataset and dropping any rows from tripids_geoids that do not match.

```
#Import Data
tripid_geoids<- haven::read_dta(file.path(spacepanels_directory,"just_ports_2022_01_26.dta"))
#Could do all this in one step, but ...
# Pick the cols that start with previous
prev<-tripid_geoids[grepl("^previous_", colnames(tripid_geoids))]
#pick the rest of the cols
tripid_geoids<-tripid_geoids[c("tripid","geoid", "namelsad", "state_fips","port_lat","port_lon")]
#cbind the two together
tripid_geoids<-cbind(tripid_geoids,prev)

APSD_DMIS_2<-merge(APSD_DMIS_2,tripid_geoids, by.x="DOCID", by.y="tripid", all.x=TRUE, all.y=FALSE)

#save to RDS
APSD_DMIS_2_name <-paste0("APSD_DMIS_2_",vintage_string,".Rds")
saveRDS(APSD_DMIS_2, file=here("data","intermediate", APSD_DMIS_2_name))
```

This match should be pretty good for source=DMIS. But it will not work for source=SFCLAM.

Loading Data fom Oracle

The APSD_DMIS_2 table must be supplemented with additional data. This section queries the Oracle databases to extract additional information.

VTR Data

Some Trip-level data in the VTR schema is needed. See table at the top. We extract them here.

```
oracle_server = "sole"
ODBC.CONNECTION <- RODBC::odbcConnect(dsn=oracle_server, uid=oracle_username, pwd=oracle_password, beli
START.YEAR = 2007
END.YEAR = 2019
RESULT.COMPILED<-list()
t<-1
for(i in START.YEAR:END.YEAR) {
  print(i)
  CURRENT.QUERY = paste("SELECT VTR.veslog",i,"t.TRIPID,tripcatg, operator, opernum, permit, nsubtrip, 
                FROM VTR.veslog",i,"t", sep="")
  RESULT.COMPILED[[t]] = sqlQuery(ODBC.CONNECTION, CURRENT.QUERY)
  t<-t+1
}

RESULT.COMPILED<-do.call(rbind.data.frame, RESULT.COMPILED)
#save to RDS
RESULT.COMPILED_name <-paste0("RESULT_COMPILED_",vintage_string,".Rds")
saveRDS(RESULT.COMPILED, file=here("data","intermediate",RESULT.COMPILED_name))
```

Scallop LA IFQ Linking variables

We also extract the activity code from DMIS. This will describe the type of trip that the vessel has declared into. The most important types of trips will be Scallop Trips; however, fishing vessels with the proper permits are allowed to retain scallops while declared into other fisheries. When this happens, the volume of scallops will be much lower.

We also extract the T/F variables corresponding the the PLAN_CAT in DMIS.

```
oracle_server = "sole"
CURRENT.QUERY = paste ("SELECT TRIP_ID, DOCID, ACTIVITY_CODE, LGC_A, LGC_B, LGC_C, SC_2, SC_3, SC_4, SC
 FROM APSD.t_ssb_trip_current@garfo_nefsc")
Scallop_Linkingorg = sqlQuery(ODBC.CONNECTION, CURRENT.QUERY)

odbcCloseAll()



#save to RDS
Scallop_Linkingorg_name <-paste0("Scallop_Linkingorg_",vintage_string,".Rds")
saveRDS(Scallop_Linkingorg, file=here("data","intermediate",Scallop_Linkingorg_name))
```

Add in cost data

These data should reference the "Estimation of Commercial Fishing Trip Costs Using Sea Sampling Data" paper by Samantha Werner & Geret DePiper. We will likely use the Winsorized trip cost estimates.

```
#Import Data
X2007_2012 <- read_excel(file.path(cost_directory,"2007_2012.xlsx"))
X2013_2020 <- read_excel(file.path(cost_directory,"2013_2020.xlsx"))
```

```r
#Merge 2007-2012 costs with 2013-2020 costs
all_yrs_costs <- merge(X2007_2012,X2013_2020, all = TRUE)



#save to RDS
all_yrs_costs_name <-paste0("all_yrs_costs_",vintage_string,".Rds")
saveRDS(all_yrs_costs, file=here("data","intermediate",all_yrs_costs_name))
```