Testing a very basic function in R

Default chunk options

Required libraries

```
library(RPostgreSQL)
```

```
## Loading required package: DBI
```

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------- tidyver
```

```
## ✓ ggplot2 2.2.1     ✓ purrr   0.2.4
## ✓ tibble  1.4.1     ✓ dplyr   0.7.4
## ✓ tidyr   0.7.2     ✓ stringr 1.2.0
## ✓ readr   1.1.1     ✓ forcats 0.2.0
```

```
## -- Conflicts ------------------------------------------------------------------------- tidyverse_cor
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
library(rjson)
library(DBI)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

Get the auxiliary data

```
source("get_HMRC_aux_data.R")
list1 <- get_HMRC_aux_data()
comcode <- data.frame(Reduce(rbind, list1[1]))
port    <- data.frame(Reduce(rbind, list1[2]))
country <- data.frame(Reduce(rbind, list1[3]))
write.csv(comcode,file="comcode.csv")
write.csv(country,file="country.csv")
```

Figure 1: A sea cucumber in all its glory. This creature kills hundreds of people every year.

Find the comcodes for

- Chicken
- Beef
- Cucumbers (watch out, beacuse there are *sea cucumbers*!)

```
cc_chicken      <- comcode[grep('CHICKEN',   toupper(comcode$description)),]
cc_all_cucumber <- comcode[grep('CUCUMBER',  toupper(comcode$description)),]
cc_cucumber     <- cc_all_cucumber[grep('VEGETABLES',toupper(cc_all_cucumber$description)),]
cc_beef         <- comcode[grep('BEEF',      toupper(comcode$description)),]
```

This is Warren's magic with a little bit of extra work

Is ten minutes too long? Then load the csv files written at the end of this notebook

```
source("get_Comtrade_data.R")
stime <- Sys.time()
polish_chicken   <- get_Comtrade_data(201001,201601,"default","02071","616")
spanish_cucumber <- get_Comtrade_data(201001,201601,"default","2001","724")
brazilian_beef   <- get_Comtrade_data(201001,201601,"default","16025","76")
etime <- Sys.time()
(etime-stime)
```

```
## Time difference of 9.194324 mins
```

Get the price in usd per kilogram

```
polish_chicken   <- polish_chicken   %>% mutate(price_kg_usd = trade_value_usd/netweight_kg)
spanish_cucumber <- spanish_cucumber %>% mutate(price_kg_usd = trade_value_usd/netweight_kg)
brazilian_beef   <- brazilian_beef   %>% mutate(price_kg_usd = trade_value_usd/netweight_kg)
```

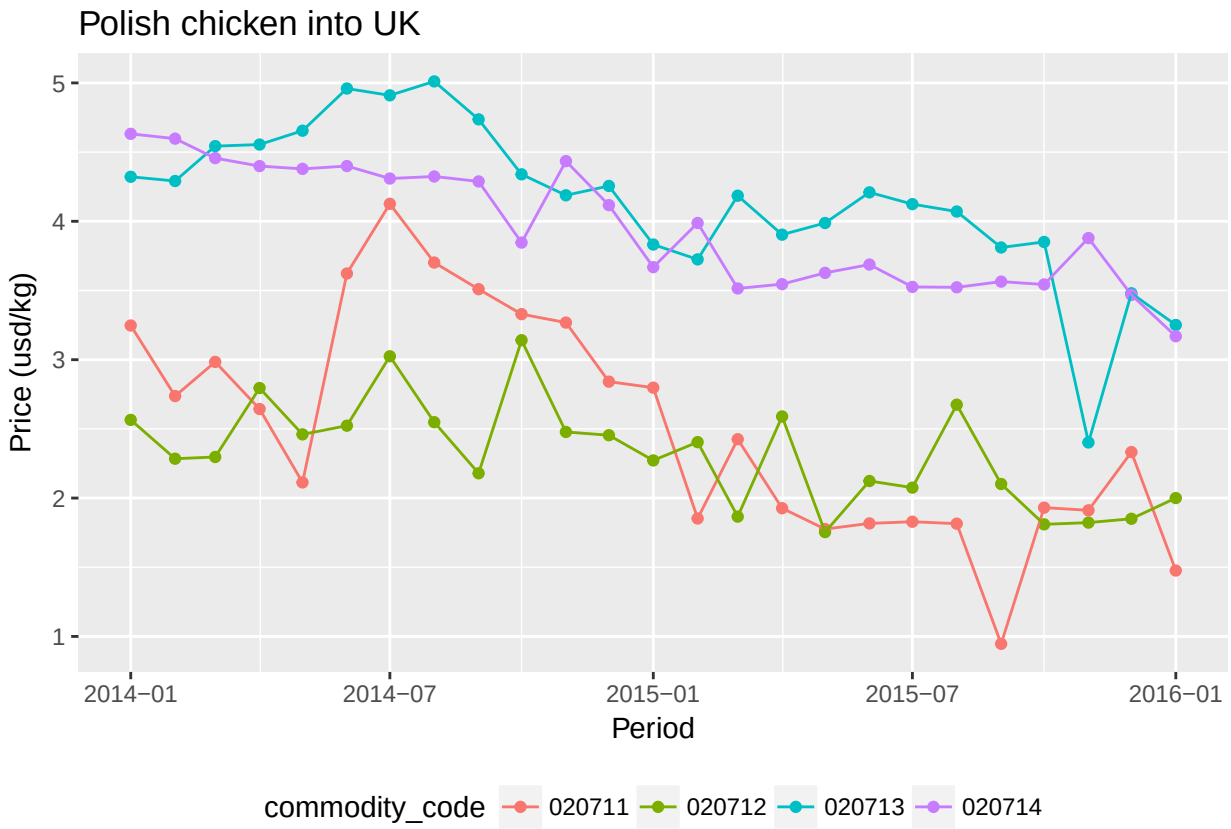
Refurbish the date into something R understand

```
polish_chicken   <- polish_chicken %>%
                    mutate(period_date = ymd(paste(period,"01",sep=""))) %>%
                    select(-period)
spanish_cucumber <- spanish_cucumber %>%
                    mutate(period_date = ymd(paste(period,"01",sep=""))) %>%
                    select(-period)
brazilian_beef   <- brazilian_beef %>%
                    mutate(period_date = ymd(paste(period,"01",sep=""))) %>%
                    select(-period)
```

Do some plots...

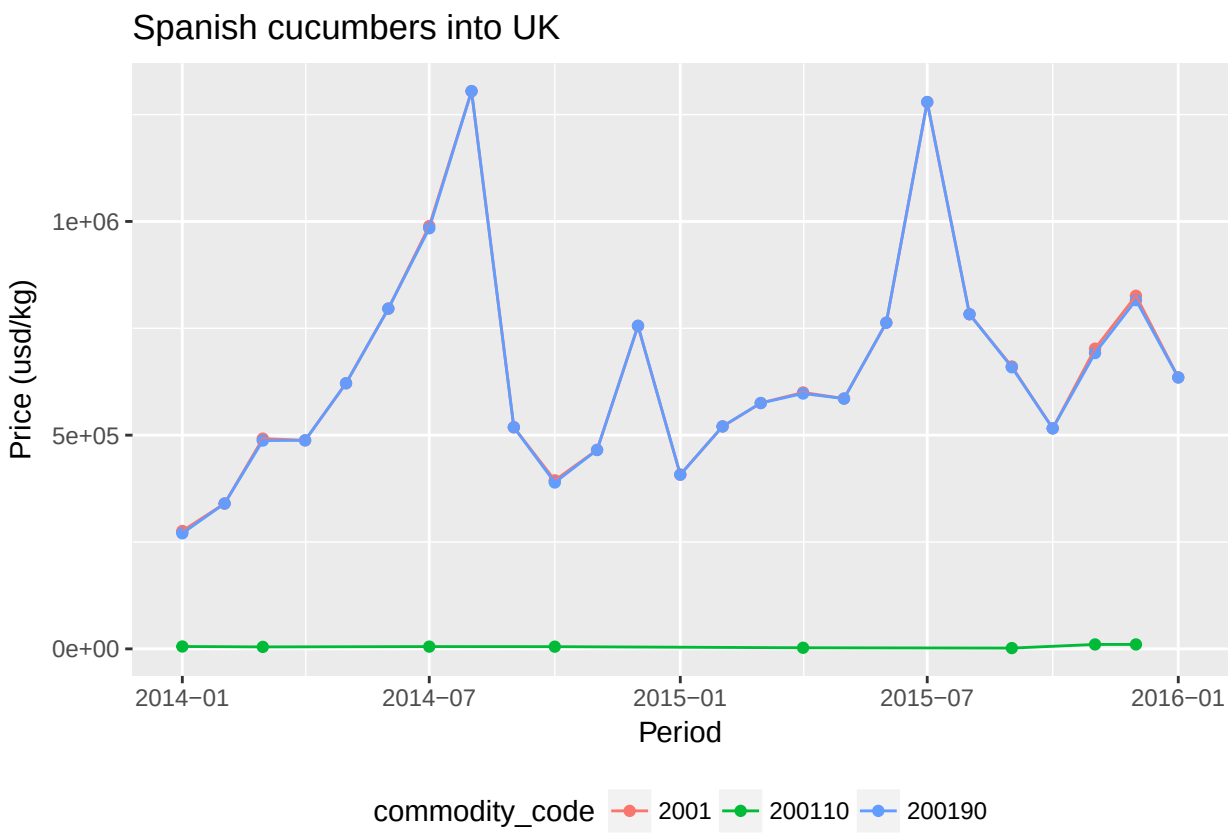
Polish chicken

```
tmp1 <- polish_chicken %>% filter(reporter=="United Kingdom") %>%
                           filter(trade_flow=="Imports")
ggplot(data=tmp1,aes(x=period_date,y=price_kg_usd,group=commodity_code,color=commodity_code)) +
  geom_point() + geom_line() +
  labs(x="Period",y="Price (usd/kg)",title="Polish chicken into UK") +
  theme(legend.position="bottom")
```



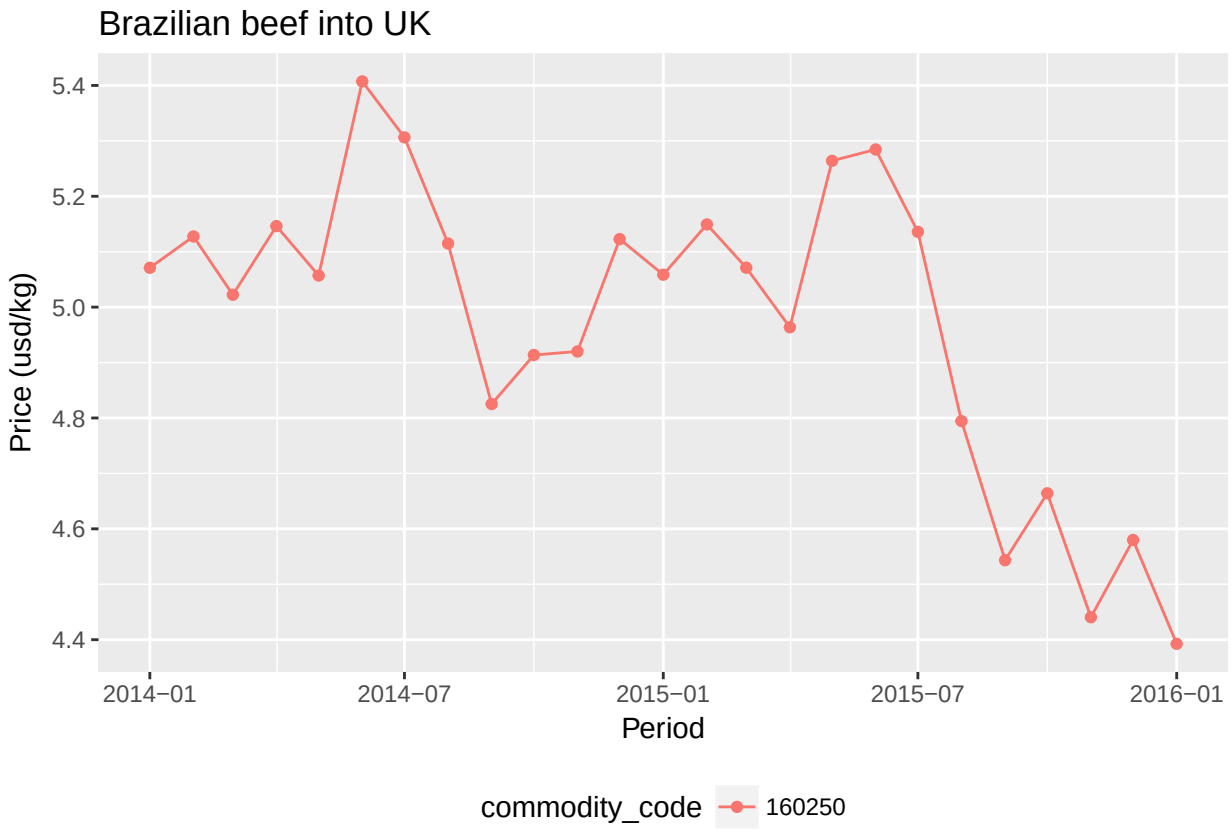
What about spanish cucumbers?

```
tmp2 <- spanish_cucumber %>% filter(reporter=="United Kingdom") %>%
                             filter(trade_flow=="Imports")
ggplot(data=tmp2,aes(x=period_date,y=netweight_kg,group=commodity_code,color=commodity_code)) +
  geom_point() + geom_line() +
  labs(x="Period",y="Price (usd/kg)",title="Spanish cucumbers into UK") +
  theme(legend.position="bottom")
```



And the brazilian beef?

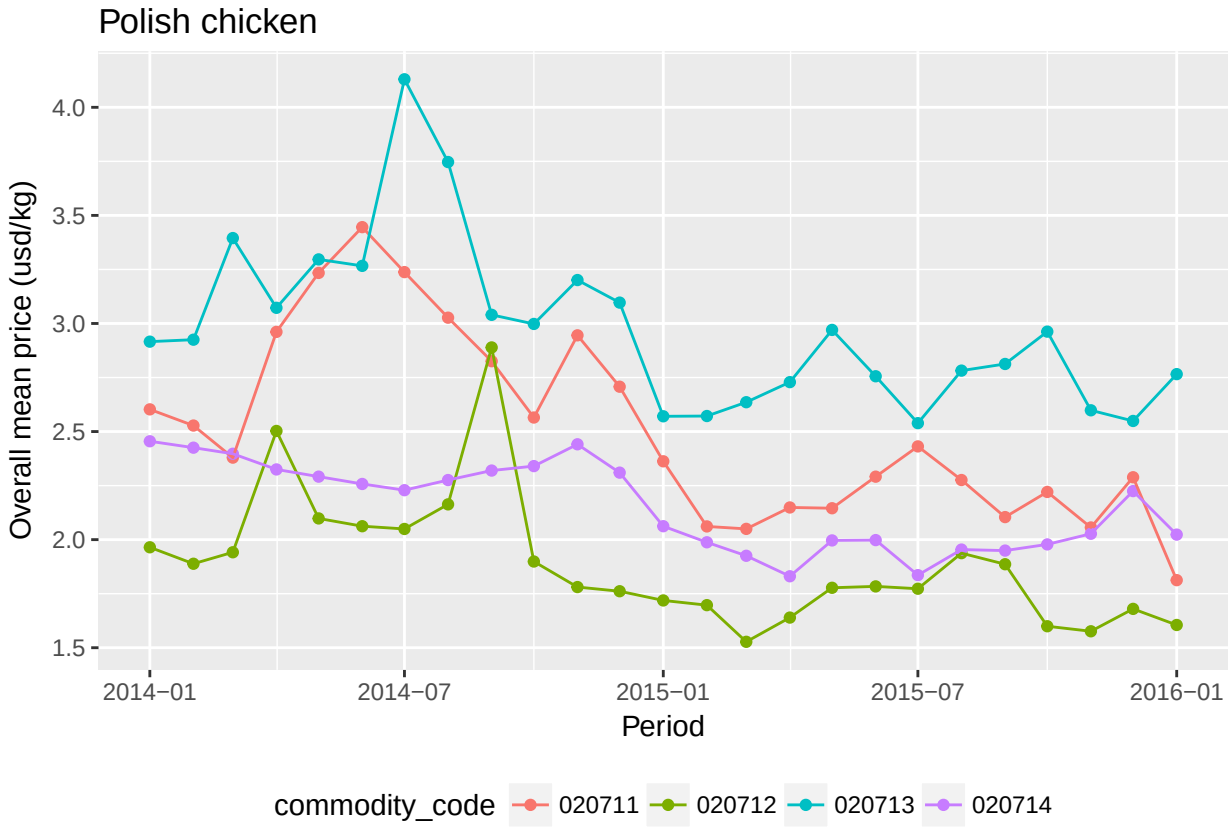
```
tmp3 <- brazilian_beef %>% filter(reporter=="United Kingdom") %>%
                           filter(trade_flow=="Imports")
ggplot(data=tmp3,aes(x=period_date,y=price_kg_usd,group=commodity_code,color=commodity_code)) +
  geom_point() + geom_line() +
  labs(x="Period",y="Price (usd/kg)",title="Brazilian beef into UK") +
  theme(legend.position="bottom")
```



Overall (mean over countries) results

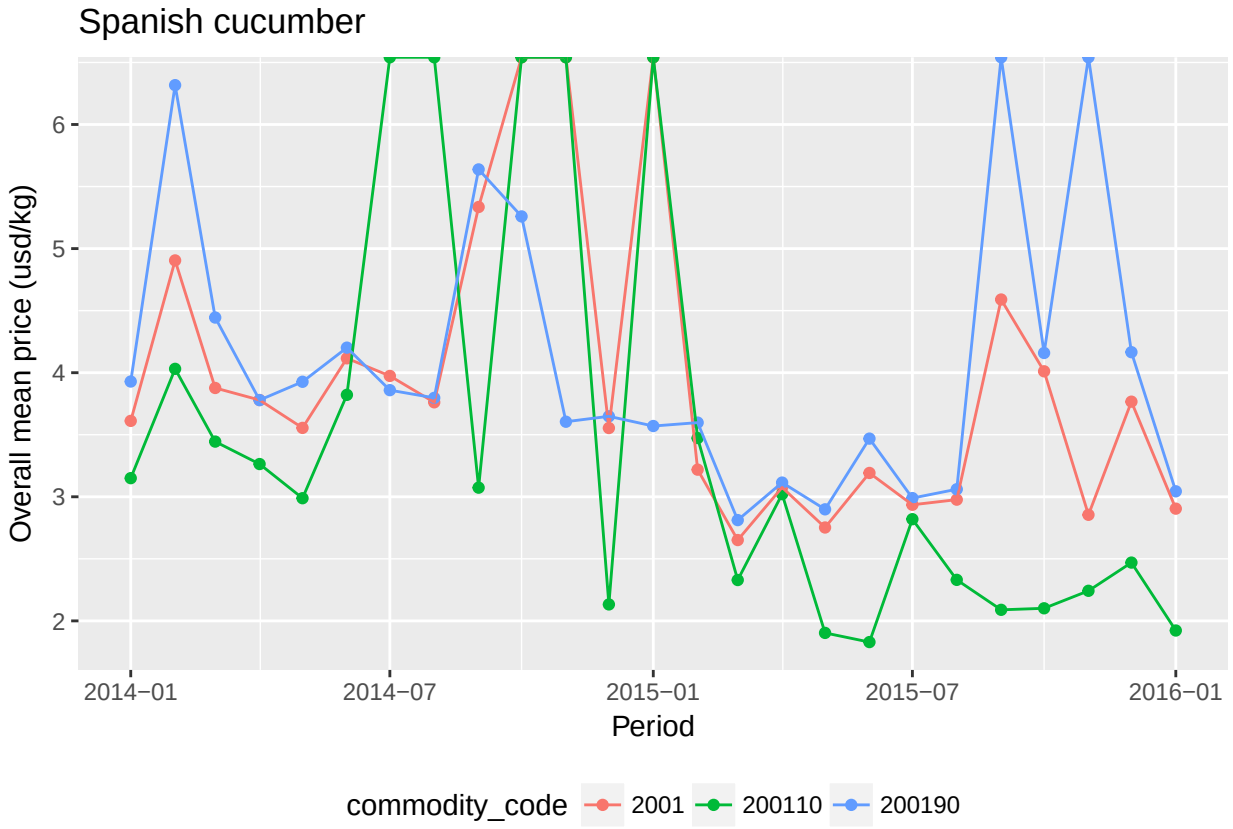
Polish chicken

```
tmp11 <- polish_chicken %>% filter(trade_flow=="Imports") %>%
                            group_by(period_date,commodity_code) %>%
                            summarize(overall_mean_price = mean(price_kg_usd,na.rm=TRUE))
ggplot(data=tmp11,aes(x=period_date,y=overall_mean_price,group=commodity_code,color=commodity_code)) +
  geom_point() + geom_line() +
  labs(x="Period",y="Overall mean price (usd/kg)",title="Polish chicken") +
  theme(legend.position="bottom")
```



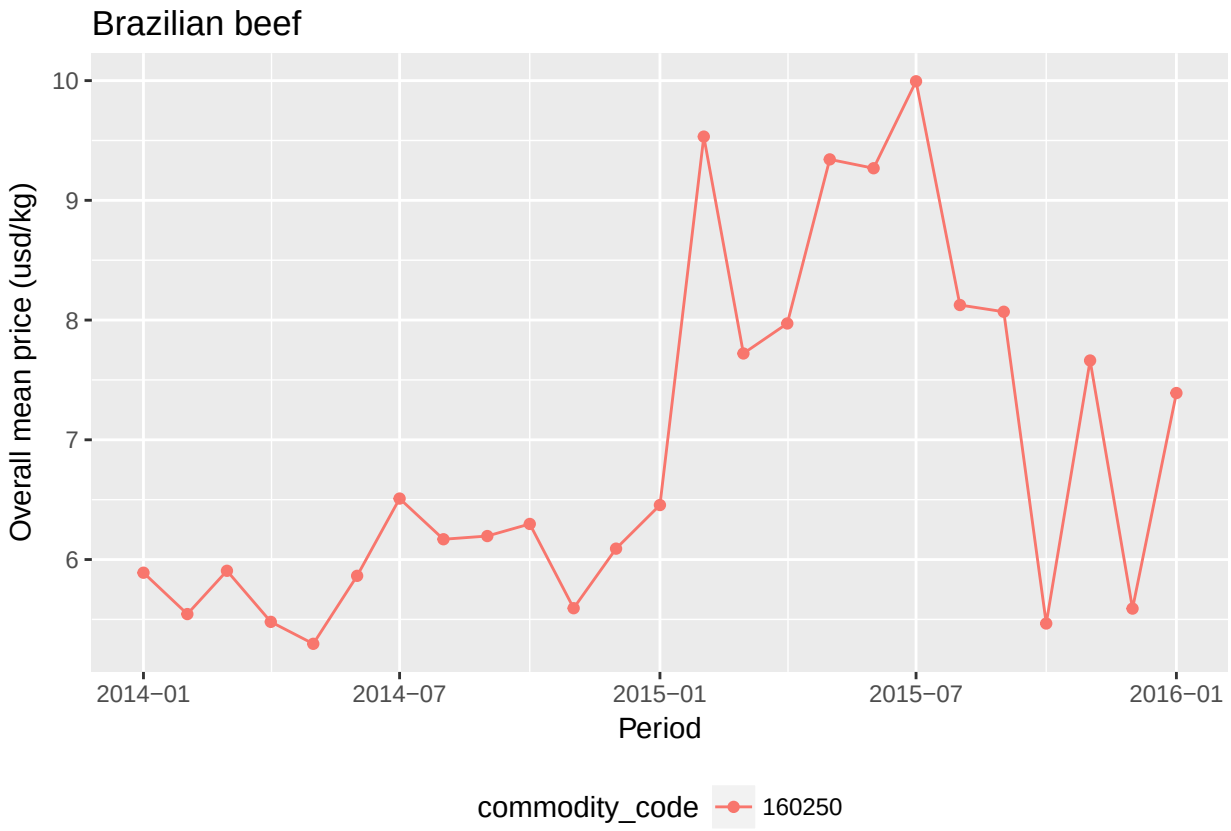
Spanish cucumbers

```
tmp22 <- spanish_cucumber %>% filter(trade_flow=="Imports") %>%
                              group_by(period_date,commodity_code) %>%
                              summarize(overall_mean_price = mean(price_kg_usd,na.rm=TRUE))
ggplot(data=tmp22,aes(x=period_date,y=overall_mean_price,group=commodity_code,color=commodity_code)) +
  geom_point() + geom_line() +
  labs(x="Period",y="Overall mean price (usd/kg)",title="Spanish cucumber") +
  theme(legend.position="bottom")
```



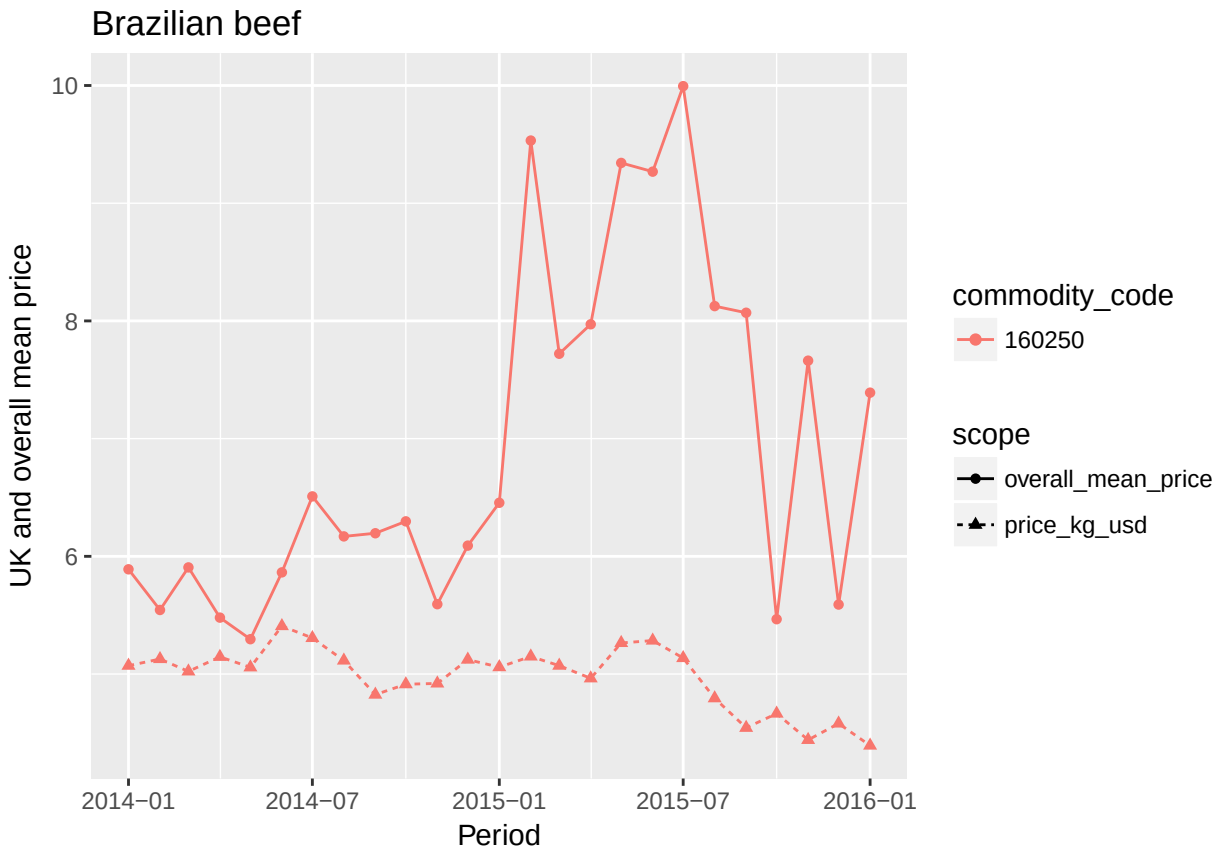
Brazilian beef

```
tmp33 <- brazilian_beef %>% filter(trade_flow=="Imports") %>%
                            group_by(period_date,commodity_code) %>%
                            summarize(overall_mean_price = mean(price_kg_usd,na.rm=TRUE))
ggplot(data=tmp33,aes(x=period_date,y=overall_mean_price,group=commodity_code,color=commodity_code)) +
  geom_point() + geom_line() +
  labs(x="Period",y="Overall mean price (usd/kg)",title="Brazilian beef") +
  theme(legend.position="bottom")
```



Let's be bold and plot the UK and overall mean price together

```
pp1 <- inner_join(tmp3,tmp33,by=c("period_date","commodity_code")) %>% select(period_date,commodity_cod
ggplot(data=pp1) + geom_line(mapping = aes(x=period_date,y=price,
                                           group=interaction(commodity_code,scope),
                                           color=commodity_code,linetype=scope)) +
                   geom_point(mapping = aes(x=period_date,y=price,
                                           group=interaction(commodity_code,scope),
                                           color=commodity_code,shape=scope)) +
                   labs(x="Period",y="UK and overall mean price",title="Brazilian beef")
```



Dump the data into csv files

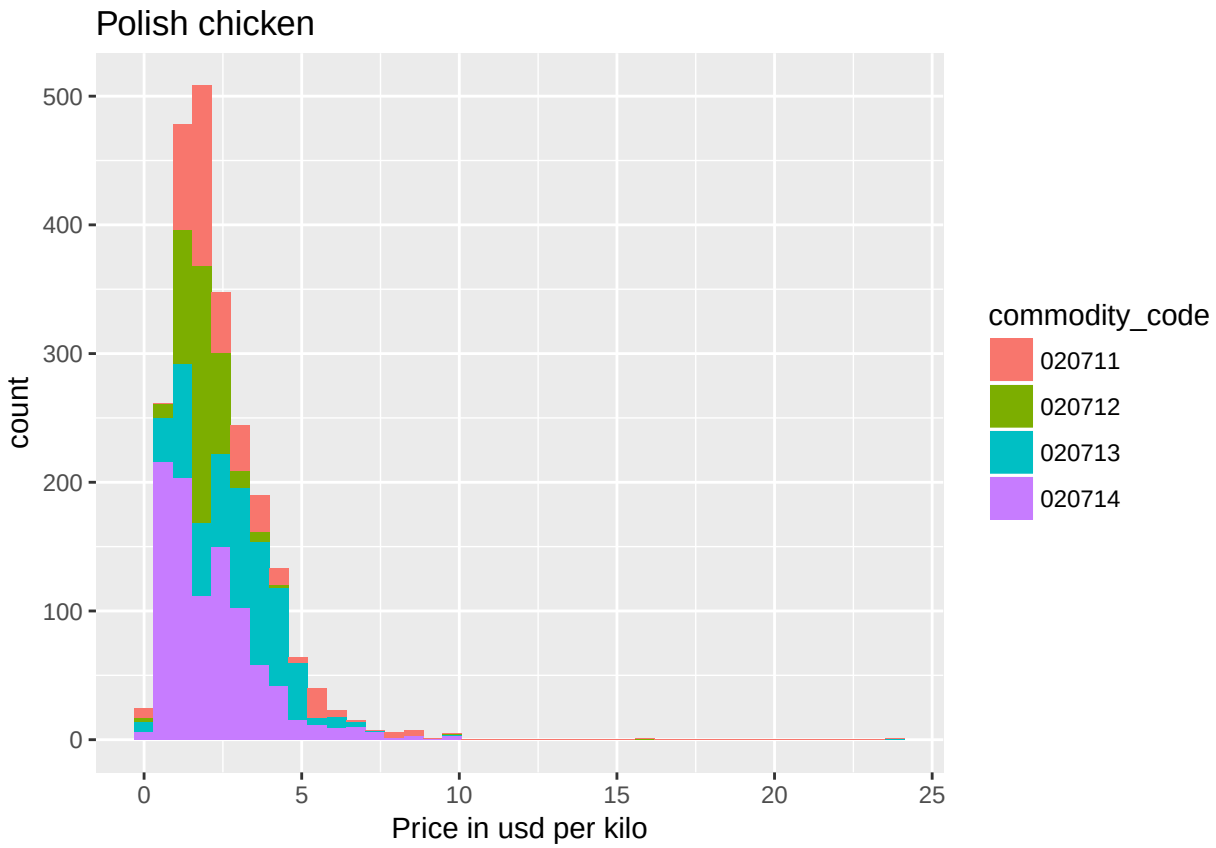
```
write.csv(polish_chicken,   file="polish_chicken.csv")
write.csv(spanish_cucumber, file="spanish_cucumber.csv")
write.csv(brazilian_beef,   file="brazilian_beef.csv")
```

Plot the pdf's

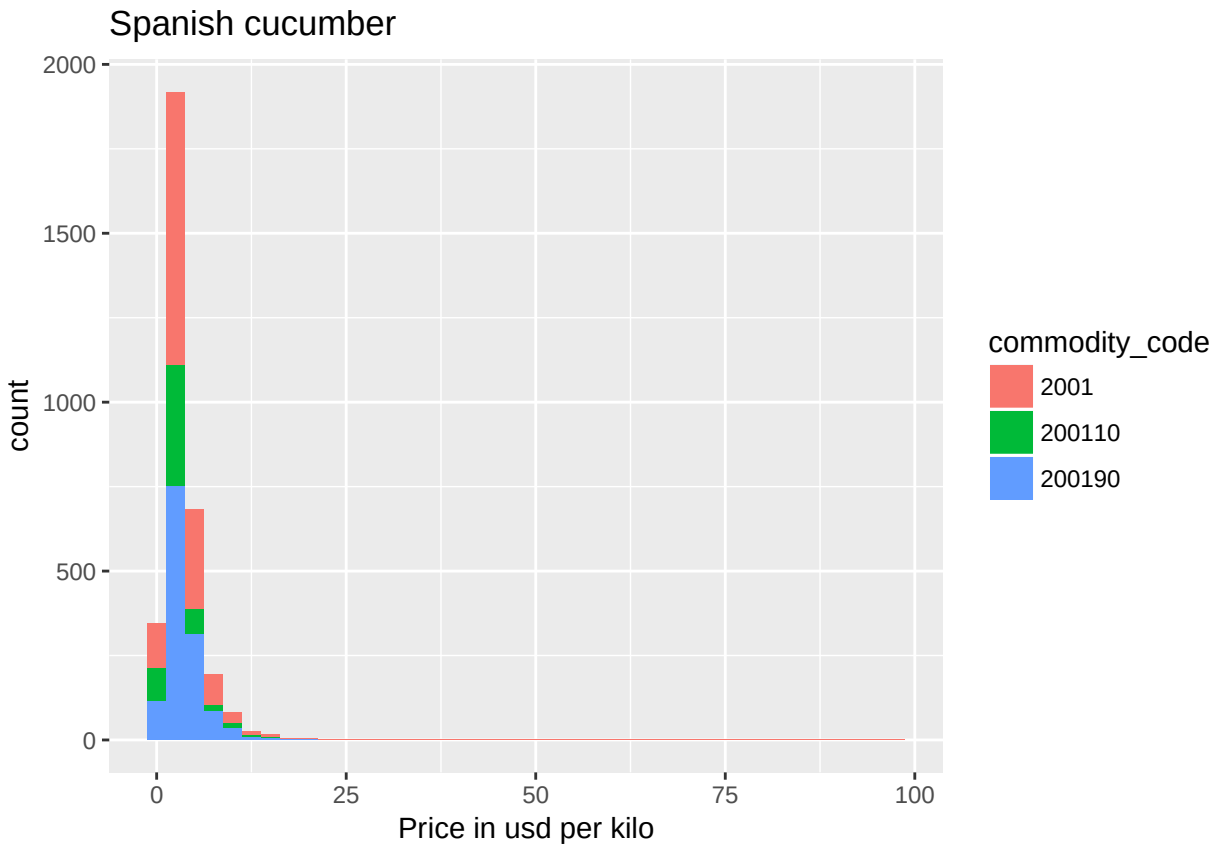
Is this a β inverted distribution?

What are those outliers? Double check the whole thing...

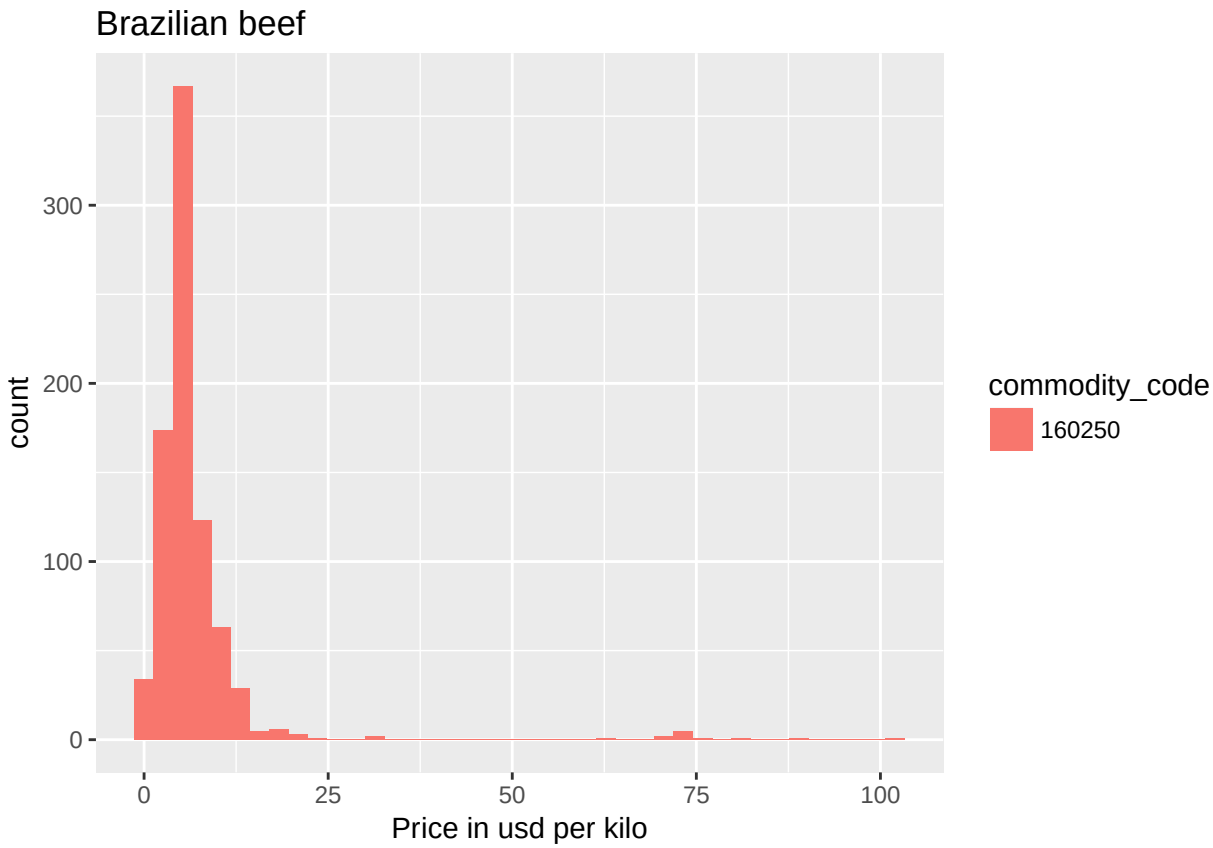
```
ggplot(data=polish_chicken %>% filter(trade_flow=="Imports") %>% group_by(commodity_code)) +
  geom_histogram(aes(price_kg_usd,fill=commodity_code),bins=40) +
  labs(x="Price in usd per kilo",title="Polish chicken")
```




```
ggplot(data=spanish_cucumber %>% filter(trade_flow=="Imports") %>% group_by(commodity_code)) +
  geom_histogram(aes(price_kg_usd,fill=commodity_code),bins=40)+
  labs(x="Price in usd per kilo",title="Spanish cucumber")
```




```
ggplot(data=brazilian_beef %>% filter(trade_flow=="Imports") %>% group_by(commodity_code)) +
  geom_histogram(aes(price_kg_usd,fill=commodity_code),bins=40)+
  labs(x="Price in usd per kilo",title="Brazilian beef")
```


Brazilian beef
count
300
200
100
0
0
25
50
75
100
Price in usd per kilo
commodity_code
160250