Stardew Valley Fish Analysis

Christina Sauro

2023-04-05

Introduction

Stardew Valley is a popular farming simulator video game that is available on Steam. The game revolves around the user inheriting their grandfather's farm in an area called Pelican Town. There are several gameplay possibilities in the simulator, including cooking, foraging, crafting, fishing, farming, interacting with villagers of the town, and more.

The game is built on a basic economic system with a currency of Gold. Players are motivated to earn Gold to develop their farm and purchase things for themselves. A large part of the game's economy is its fishing minigame. The player can catch different types of fish with different rarities and uses for their progress in the game. This project will be an analysis of the various types of fish in Stardew Valley and focus on the following questions:

1. Do fish sell higher or lower depending on type (basic, legendary, night market, catchable)?

- Even though it's pretty evident that basic fish will not be selling the highest, it might be useful to the player to see what category of fish sells for more money, as they can focus on type of fish to catch. I would predict that legendary fish would sell for the most, as they are normally found in rare places and at specific times.

2. What are most fish used for (e.g. bundles, food, collectibles, etc.)?

- Each player can have different purposes for fishing in the game depending on their needs, so a player might focus on different fish depending on their use. I think that my results will show that most fish in the game will be used for finishing bundles, since they allow the player to advance the most in the game and unlock more items.

3. Where are most fish located among the game map?

- Depending where the user decides to fish, they can maximize the variety of fish they catch. I predict that most fish will be found in the main ocean of the town, as it's the most common area and one of the first places they are exposed to in the game.

This fish catalog data from Kaggle and can be found **here**. The villagers.csv file will not be included in this analysis.

Data Wrangling Plan

I will be cleaning the seven files separately.

Iteration 1 - fish_detail.csv

Phase 1

1. Read in the CSV file
2. Change category names to lowercase
3. Check if the data is in tidy format, pivot if the data is not
4. Idenfity uid's
5. Sort by uid's

Phase 2

```
## 1 and 2
fish_detail <- read_csv("fish_detail.csv", col_types="??????????") %>%
  rename_with(tolower)

fish_detail %>% head(5)
```

```
## # A tibble: 5 x 10
##   name      descr~1 locat~2 time  season weather size ~3 diffi~4 base ~5 used ~6
##   <chr>     <chr>   <chr>   <chr> <chr>  <chr>   <chr>   <chr>     <dbl> <chr>
## 1 Pufferfi~ Inflat~ Ocean,~ 12pm~ "Summ~ Sun     1-37    80 flo~      29 "Speci~
## 2 Anchovy   A smal~ Ocean   Anyt~ "Spri~ Any     1-17    30 dart      13 "No Us~
## 3 Tuna      A larg~ Ocean,~ 6am ~ "Summ~ Any     12-61   70 smo~      26 "Ocean~
## 4 Sardine   A comm~ Ocean   6am ~ "Spri~ Any     1-13    30 dart      13 "Ocean~
## 5 Bream     A fair~ River ~ 6pm ~ "All ~ Any     12-31   35 smo~      14 "Night~
## # ... with abbreviated variable names 1: description, 2: location,
## #   3: `size (inches)`, 4: `difficulty & behavior`, 5: `base xp`, 6: `used in`
```

3. I believe this data is tidy as a unique observation for each fish is noted in each row.

The uid in this set would be the name of the fish, as it is the identifier of the observations in each row. To check this, I can use the following code chunk:

```
## 4
fish_detail %>%
  count(name, location, season, weather) %>%
  filter(n > 1) %>%
  dim_desc()
```

```
## [1] "[0 x 5]"
```

I confirmed that the fish name is the uid.

```
## 5
fish_detail %<>%
  arrange(name)
```

Iteration 2- fish_detail.csv

Phase 1

1. Convert character columns to lowercase
2. Separate the difficulty & behaviour column
3. Add new generalized columns

- Add new column for location, use, and type

4. Convert any character columns to factors

Phase 2

```
## 1

fish_detail %<>%
  mutate(across(c("name", "description", "location", "season", "used in", "weather"), tolower))
```

```
## 2
fish_detail %<>%
  separate("difficulty & behavior", c("difficulty", "behaviour"), sep = " ", convert=TRUE)
```

For the location and usage columns, I used str_detect to check for seventeen common locations and five usages I noticed when looking through the data.

```
## 3
fish_detail %<>%
  mutate( located = case_when(
    location %>% str_detect("ginger island") ~ "ginger island",
    location %>% str_detect("mines") ~ "mines",
    location %>% str_detect("mountain lake") ~ "mountain lake",
    location %>% str_detect("river") ~ "river",
    location %>% str_detect("ocean") ~ "ocean",
    location %>% str_detect("witch's swamp") ~ "witch's swamp",
    location %>% str_detect("forest ") ~ "forest",
    location %>% str_detect("secret woods") ~ "secret woods",
    location %>% str_detect("mutant bug lair") ~ "mutant bug lair",
    location %>% str_detect("desert") ~ "desert",
    location %>% str_detect("volcano caldera") ~ "volcano caldera",
    location %>% str_detect("the sewers") ~ "the sewers",
    location %>% str_detect("jojamart") ~ "jojamart",
    location %>% str_detect("night market") ~ "night market",
    location %>% str_detect("freshwater") ~ "freshwater",
    location %>% str_detect("beach") ~ "beach",
    location %>% str_detect("any") ~ "any"
  )
  ) %>%
  select(name, location, located, everything())
```

I used the same method to create a new "usage" column. To check if a fish is being used for a food recipe, I created a vector of fish dishes from the game and used str_detect() to see if the fields matched any of the entries. The collected fish recipes are from the **Stardew Valley Wiki** and can be found at the connected link.

```
fish_recipes <- c("baked fish", "fried calamari", "lucky lunch", "carp surprise", "salmon dinner",
                  "fish taco", "crispy bass", "tom kha soup", "trout soup", "fried eel", "spicy eel",
                  "sashimi", "maki roll", "dish o' the sea", "seafoam pudding", "algae soup",
                  "pale broth", "chowder", "fish stew", "lobster bisque", "crab cakes",
                  "shrimp cocktail")

fish_detail %<>%
  mutate(
    usage = (
      case_when(
        `used in` %>% str_detect("quest") ~ "quest",
        `used in` %>% str_detect("(loved gift)") ~ "gifts for villagers",
        `used in` %>% str_detect("bundle") ~ "bundle",
        `used in` %>% str_detect("no uses") ~ "no uses",
        `used in` %>% str_detect(str_c(fish_recipes, collapse = "|")) ~ "food",
      )
    )
  ) %>%
  glimpse()
```

```
## Rows: 47
## Columns: 13
## $ name            <chr> "albacore", "anchovy", "blue discus", "bream", "bullhe~
## $ location        <chr> "ocean", "ocean", "ginger island pond and rivers", "ri~
## $ located         <chr> "ocean", "ocean", "ginger island", "river", "mountain ~
## $ description     <chr> "prefers temperature \"edges\" where cool and warm wat~
## $ time            <chr> "6am - 11am\n6pm - 2am", "Anytime", "Anytime", "6pm - ~
## $ season          <chr> "fall\n winter", "spring\n fall", "all seasons", "all ~
## $ weather         <chr> "any", "any", "any", "any", "any", "any", "rain", "any~
## $ `size (inches)` <chr> "20-41", "1-17", "2-10", "12-31", "12-31", "15-51", "1~
## $ difficulty      <int> 60, 30, 60, 35, 46, 15, 75, 35, 78, 70, 50, 50, 50, 25~
## $ behaviour       <chr> "mixed", "dart", "dart", "smooth", "smooth", "mixed", ~
## $ `base xp`       <dbl> 23, 13, 23, 14, 18, 8, 28, 14, 29, 26, 19, 19, 19, 11,~
## $ `used in`       <chr> "\"fish stew\" quest", "no uses", "no uses", "night fi~
## $ usage           <chr> "quest", "no uses", "no uses", "bundle", "bundle", "bu~
```

```
fish_detail %<>%
  mutate(
    type = "basic"
  )

fish_detail %<>% select(!c(location, `used in`)) # Dropping the columns that were used for recoding
```

```
## 4
fish_detail %<>%
  mutate(across(!c("difficulty", "base xp"), as.factor))
```

Iteration 1 - fish_price_breakdown.csv

Phase 1

1. Read in the CSV file
2. Change category names to lowercase
3. Check if the data is in tidy format, pivot if the data is not
4. Idenfity uid's
5. Sort by uid's

Phase 2

```
## 1 and 2
fish_breakdown <- read_csv("fish_price_breakdown.csv", col_types=paste(rep("c", 48), collapse=""))

fish_breakdown %<>%
  rename_with(tolower)

fish_breakdown %>% head(5)
```

```
## # A tibble: 5 x 48
##   name        puffe~1 anchovy tuna  sardine bream large~2 small~3 rainb~4 salmon
##   <chr>       <chr>   <chr>   <chr> <chr>   <chr> <chr>   <chr>   <chr>   <chr>
## 1 Base Price  200g    30g     100g  40g     45g   100g    50g     65g     75g
## 2 BP Silver   250g    37g     125g  50g     56g   125g    62g     81g     93g
## 3 BP Gold     300g    45g     150g  60g     67g   150g    75g     97g     112g
## 4 BP Irridium 400g    60g     200g  80g     90g   200g    100g    130g    150g
## 5 Fisher Pro~ 250g    37g     125g  50g     56g   125g    62g     81g     93g
## # ... with 38 more variables: walleye <chr>, perch <chr>, carp <chr>,
## #   catfish <chr>, pike <chr>, sunfish <chr>, `red mullet` <chr>,
## #   herring <chr>, eel <chr>, octopus <chr>, `red snapper` <chr>, squid <chr>,
## #   `sea cucumber` <chr>, `super cucumber` <chr>, ghostfish <chr>,
## #   stonefish <chr>, `ice pip` <chr>, `lava eel` <chr>, sandfish <chr>,
## #   `scorpion carp` <chr>, flounder <chr>, `midnight carp` <chr>,
## #   sturgeon <chr>, `tiger trout` <chr>, bullhead <chr>, tilapia <chr>, ...
```

3. This data is not tidy and will need to be pivoted. Fish can be uniquely identified by their names, so I will pivot based on that category.

```
## 3
fish_breakdown %<>%
  pivot_longer(!c("name"), names_to = "fish name", values_to = "price")
```

```
## 4 - checking that fish name is the uid
fish_breakdown %>%
  count(`fish name`, name, price) %>%
  filter(n > 1) %>%
  dim_desc()
```

```
## [1] "[0 x 4]"
```

I've confirmed that the data is tidy and fish name is the uid.

```
## 5
fish_breakdown %<>%
  select(`fish name`, everything())
fish_breakdown %<>%
  arrange(`fish name`)
```

Iteration 2- fish_price_breakdown.csv

Phase 1

1. Convert character columns to lowercase
2. Create classification column
3. Recode name column & rename it to rank

- The prefixes "bp", "fp", and "ap" will need to be removed.

4. Recode rank column

- Iridium was incorrectly spelled as irridium and needs to be changed

5. Recode price column

- Each entry has a "g" at the end of the value, meaning the gold currency. This should be removed to plot this column easier.

6. Convert price column to a numeric column
7. Convert any character columns to factors
8. After cleaning the other tibbles, join the data into one tibble

- Pivot all tibbles wider
- Join regular fish and legendary fish tibbles together separately
- Add rows of each pivoted tibble based on fish name

Phase 2

```
## 1
fish_breakdown %<>%
  mutate(name = tolower(name))
```

Similar to how I checked the location column in the fish_detail data wrangling plan, I used a similar method to check the classification of the fish. Depending on the job of the player (no job, fisher, angler), some fish can sell for more Gold.

```r
## 2
fish_breakdown %<>%
  mutate(
    classification = (
      case_when(
        name %>% str_detect("bp|base price") ~ "base price",
        name %>% str_detect("fp|fisher profession") ~ "fisher profession",
        name %>% str_detect("ap|angler profession") ~ "angler profession"
      )
    )
  ) %>%
  select(name, classification, everything())

fish_breakdown %>% head(5)
```

```
## # A tibble: 5 x 4
##   name                      classification    `fish name` price
##   <chr>                     <chr>             <chr>       <chr>
## 1 base price                base price        albacore    75g
## 2 bp silver                 base price        albacore    93g
## 3 bp gold                   base price        albacore    112g
## 4 bp irridium               base price        albacore    150g
## 5 fisher profession (+25%) fisher profession albacore    93g
```

```r
## 3
fish_breakdown %<>%
  mutate(name = str_remove_all(name, "bp |fp |ap "))

fish_breakdown %<>%
  mutate(name = case_when(
    str_detect(name, "base price|fisher profession|angler profession") ~ "regular",
    TRUE ~ name
          )
           )

fish_breakdown %<>%
  rename(rank = name)

fish_breakdown %<>% # renaming `fish name` to name to make future joining easier
  rename(name = `fish name`)
```

```r
## 4
fish_breakdown %<>%
  mutate(rank = str_replace_all(rank, "irridium", "iridium"))
```

```r
## 5
fish_breakdown %<>%
  mutate(price =  price %>% str_remove_all("g|,"))

fish_breakdown %>% head(5) #checking that all the recoding worked
```

```
## # A tibble: 5 x 4
```

```
##   rank    classification    name     price
##   <chr>   <chr>             <chr>    <chr>
## 1 regular base price        albacore 75
## 2 silver  base price        albacore 93
## 3 gold    base price        albacore 112
## 4 iridium base price        albacore 150
## 5 regular fisher profession albacore 93
```

```
## 6
fish_breakdown %<>%
  mutate(price = as.numeric(price))

fish_breakdown %<>%
  select(name, everything())
```

```
## 7
fish_breakdown %<>%
  mutate(across(name:classification, as.factor))
```

I used the same process in the data wrangling plans to clean the rest of the files to be joined:
- legendary_fish_detail.csv
- legendary_fish_price_breakdown.csv
- legendaryfishII.csv
- nightmarketfish.csv
- crabpotandothercatchables.csv

```
## New names:
## * `` -> `...8`
```

```
legendary_fish_detail %>%
  head(5)
```

```
## # A tibble: 5 x 13
##   name      located descr~1 locat~2 time  season weather size ~3 diffi~4 behav~5
##   <fct>     <fct>   <fct>   <fct>   <fct> <fct>  <fct>   <fct>     <int> <fct>
## 1 angler    jojama~ uses a~ north ~ any   fall   any     17-19        85 smooth
## 2 crimsonf~ beach   lives ~ east p~ any   summer any     19-21        95 mixed
## 3 glacierf~ forest  builds~ south ~ any   winter any     26-28       100 mixed
## 4 legend    mounta~ the ki~ the mo~ any   spring rain    49-51       110 mixed
## 5 mutant c~ the se~ the st~ the se~ any   all s~ any     35-37        80 dart
## # ... with 3 more variables: `base xp` <dbl>, usage <fct>, type <fct>, and
## #   abbreviated variable names 1: description, 2: location, 3: `size (inches)`,
## #   4: difficulty, 5: behaviour
```

```
legendary_fish_breakdown %>%
  head(5)
```

```
## # A tibble: 5 x 4
##   name   rank    classification    price
##   <fct>  <fct>   <fct>             <dbl>
## 1 angler regular base price          900
## 2 angler silver  base price         1125
```

```
## 3 angler gold    base price         1350
## 4 angler iridium base price         1800
## 5 angler regular fisher profession  1350
```

```
legendary_fish_ii %>%
  head(5)
```

```
## # A tibble: 5 x 16
##   name        rank  class~1 located descr~2 locat~3 time  season weather size ~4
##   <fct>       <fct> <fct>   <fct>   <fct>   <fct>   <fct> <fct>  <fct>   <fct>
## 1 glacierfis~ regu~ base p~ forest  the or~ south ~ any   all s~ any     26-28
## 2 glacierfis~ silv~ base p~ forest  the or~ south ~ any   all s~ any     26-28
## 3 glacierfis~ gold  base p~ forest  the or~ south ~ any   all s~ any     26-28
## 4 glacierfis~ irid~ base p~ forest  the or~ south ~ any   all s~ any     26-28
## 5 glacierfis~ regu~ fisher~ forest  the or~ south ~ any   all s~ any     26-28
## # ... with 6 more variables: difficulty <int>, behaviour <fct>,
## #   `base xp` <dbl>, usage <fct>, type <fct>, price <dbl>, and abbreviated
## #   variable names 1: classification, 2: description, 3: location,
## #   4: `size (inches)`
```

```
nightmarket_fish %>%
  head(5)
```

```
## # A tibble: 5 x 13
##   name     rank    class~1 located descr~2 size  diffi~3 behav~4 base ~5 used ~6
##   <fct>    <fct>   <fct>   <fct>   <fct>   <fct>   <int> <fct>     <dbl> <fct>
## 1 blobfish regular base p~ night ~ this o~ 8-25       75 floater      28 master~
## 2 blobfish silver  base p~ night ~ this o~ 8-25       75 floater      28 master~
## 3 blobfish gold    base p~ night ~ this o~ 8-25       75 floater      28 master~
## 4 blobfish iridium base p~ night ~ this o~ 8-25       75 floater      28 master~
## 5 blobfish regular fisher~ night ~ this o~ 8-25       75 floater      28 master~
## # ... with 3 more variables: usage <fct>, type <fct>, price <dbl>, and
## #   abbreviated variable names 1: classification, 2: description,
## #   3: difficulty, 4: behaviour, 5: `base xp`, 6: `used in`
```

```
crabpot_catchables %>%
  head(5)
```

```
## # A tibble: 5 x 9
##   name  rank    classification    located description  size ~1 usage type  price
##   <fct> <fct>   <fct>             <fct>   <fct>        <fct>   <fct> <fct> <dbl>
## 1 clam  regular base price        ocean   someone liv~ 1-5     food  crab~    50
## 2 clam  silver  base price        ocean   someone liv~ 1-5     food  crab~    62
## 3 clam  gold    base price        ocean   someone liv~ 1-5     food  crab~    75
## 4 clam  iridium base price        ocean   someone liv~ 1-5     food  crab~   100
## 5 clam  regular fisher profession ocean   someone liv~ 1-5     food  crab~    50
## # ... with abbreviated variable name 1: `size (inches)`
```

```
## 8 - pivotting the basic fish tibbles to be longer
fish_breakdown %<>%
  pivot_wider(names_from = "classification", values_from = "price")
```

```
fish_breakdown %<>%
  pivot_wider(names_from = "rank", values_from = c("base price", "fisher profession",
                                                   "angler profession"), names_sep = " ")
```

I pivoted the other 5 files to be wider as well based on their classification.

```
## 8 - joining the basic and legendary fish tibbles together separately

fish_tibble <- left_join(fish_detail, fish_breakdown, by = "name")

fish_tibble %>% # checking that left join worked
  head(5)
```

```
## # A tibble: 5 x 24
##   name      located descr~1 time  season weather size ~2 diffi~3 behav~4 base ~5
##   <fct>     <fct>   <fct>   <fct> <fct>  <fct>   <fct>     <int> <fct>     <dbl>
## 1 albacore  ocean   "prefe~ "6am~ "fall~ any     20-41        60 mixed        23
## 2 anchovy   ocean   "a sma~ "Any~ "spri~ any     1-17         30 dart         13
## 3 blue dis~ ginger~ "a bri~ "Any~ "all ~ any     2-10         60 dart         23
## 4 bream     river   "a fai~ "6pm~ "all ~ any     12-31        35 smooth       14
## 5 bullhead  mounta~ "a rel~ "Any~ "all ~ any     12-31        46 smooth       18
## # ... with 14 more variables: usage <fct>, type <fct>,
## #   `base price regular` <dbl>, `base price silver` <dbl>,
## #   `base price gold` <dbl>, `base price iridium` <dbl>,
## #   `fisher profession regular` <dbl>, `fisher profession silver` <dbl>,
## #   `fisher profession gold` <dbl>, `fisher profession iridium` <dbl>,
## #   `angler profession regular` <dbl>, `angler profession silver` <dbl>,
## #   `angler profession gold` <dbl>, `angler profession iridium` <dbl>, and ...
```

```
leg_fish_tibble <- left_join(legendary_fish_detail, legendary_fish_breakdown, by = "name")
leg_fish_tibble %<>%
  bind_rows(legendary_fish_ii)

leg_fish_tibble %>% # checking that left join and binding worked
  head(5)
```

```
## # A tibble: 5 x 25
##   name      located descr~1 locat~2 time  season weather size ~3 diffi~4 behav~5
##   <fct>     <fct>   <fct>   <fct>   <fct> <fct>  <fct>   <fct>     <int> <fct>
## 1 angler    jojama~ uses a~ north ~ any   fall   any     17-19        85 smooth
## 2 crimsonf~ beach   lives ~ east p~ any   summer any     19-21        95 mixed
## 3 glacierf~ forest  builds~ south ~ any   winter any     26-28       100 mixed
## 4 legend    mounta~ the ki~ the mo~ any   spring rain    49-51       110 mixed
## 5 mutant c~ the se~ the st~ the se~ any   all s~ any     35-37        80 dart
## # ... with 15 more variables: `base xp` <dbl>, usage <fct>, type <fct>,
## #   `base price regular` <dbl>, `base price silver` <dbl>,
## #   `base price gold` <dbl>, `base price iridium` <dbl>,
## #   `fisher profession regular` <dbl>, `fisher profession silver` <dbl>,
## #   `fisher profession gold` <dbl>, `fisher profession iridium` <dbl>,
## #   `angler profession regular` <dbl>, `angler profession silver` <dbl>,
## #   `angler profession gold` <dbl>, `angler profession iridium` <dbl>, and ...
```

The legendary_fish_ii tibble contained information related to the fish's details and price, so to join it properly with the other two tibbles, I needed to add the tibble's rows to leg_fish_tibble.
Both tibbles include the new columns and rows, which confirms the left join and binding worked.

```
## 8 - binding rows to one main tibble from data with different fish types

main_tibble <- fish_tibble %>%
  bind_rows(leg_fish_tibble, nightmarket_fish, crabpot_catchables)

main_tibble %>% # check that binding worked
  glimpse()
```

```
## Rows: 73
## Columns: 27
## $ name                         <fct> albacore, anchovy, blue discus, bream, bul~
## $ located                      <fct> ocean, ocean, ginger island, river, mounta~
## $ description                  <fct> "prefers temperature \"edges\" where cool ~
## $ time                         <fct> 6am - 11am
## 6pm - 2am, Anytime, Anytime, 6pm~
## $ season                       <fct> fall
##  winter, spring
##  fall, all seasons, all~
## $ weather                      <fct> any, any, any, any, any, any, rain, any, a~
## $ `size (inches)`              <fct> 20-41, 1-17, 2-10, 12-31, 12-31, 15-51, 12~
## $ difficulty                   <int> 60, 30, 60, 35, 46, 15, 75, 35, 78, 70, 50~
## $ behaviour                    <fct> mixed, dart, dart, smooth, smooth, mixed, ~
## $ `base xp`                    <dbl> 23, 13, 23, 14, 18, 8, 28, 14, 29, 26, 19,~
## $ usage                        <fct> quest, no uses, no uses, bundle, bundle, b~
## $ type                         <fct> basic, basic, basic, basic, basic, basic, ~
## $ `base price regular`         <dbl> 75, 30, 120, 45, 75, 30, 200, 50, 100, 85,~
## $ `base price silver`          <dbl> 93, 37, 150, 56, 93, 37, 250, 62, 125, 106~
## $ `base price gold`            <dbl> 112, 45, 180, 67, 112, 45, 300, 75, 150, 1~
## $ `base price iridium`         <dbl> 150, 60, 240, 90, 150, 60, 400, 100, 200, ~
## $ `fisher profession regular`  <dbl> 93, 37, 150, 56, 93, 37, 250, 62, 125, 106~
## $ `fisher profession silver`   <dbl> 116, 46, 187, 70, 116, 46, 312, 77, 156, 1~
## $ `fisher profession gold`     <dbl> 140, 56, 225, 83, 140, 56, 375, 93, 187, 1~
## $ `fisher profession iridium`  <dbl> 187, 75, 300, 112, 187, 75, 500, 125, 250,~
## $ `angler profession regular`  <dbl> 112, 45, 180, 67, 112, 45, 300, 75, 150, 1~
## $ `angler profession silver`   <dbl> 139, 55, 225, 84, 139, 55, 375, 93, 187, 1~
## $ `angler profession gold`     <dbl> 168, 67, 270, 100, 168, 67, 450, 112, 225,~
## $ `angler profession iridium`  <dbl> 225, 90, 360, 135, 225, 90, 600, 150, 300,~
## $ location                     <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ size                         <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ `used in`                    <fct> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

The main tibble has basic, legendary, night market, and crabpot/catchable names therefore the binding operation worked.

```
## 8 - pivotting the main tibble to be longer so that it can be plotted correctly

main_tibble %<>%
  pivot_longer(cols = c(contains("profession"), contains("base price")), names_to = "rank",
               values_to="price")
```

```r
main_tibble %<>%
  mutate(across(c("located", "rank", "usage"), as.factor))
```

I can drop the "location" and "used in" column as I have already recoded these columns. I can also drop the "size" column as it's not relevant to the data and it's not recorded for the legendary, nightmarket, or crabpot data.

```r
main_tibble %<>%
  select(!c("location", "size", "used in"))

main_tibble %<>% # dropping rows with NA where prices were not recorded
  drop_na(price)
```

Results/Discussion

Question 1: Do fish sell higher or lower depending on type?

```r
p1 <- main_tibble %>%
  ggplot(aes(x = price, y = fct_reorder(str_to_title(type), price, .desc = TRUE),
             colour = fct_reorder(str_to_title(type), price, .desc = TRUE))) +
  geom_boxplot(width = 0.7) +
  labs(x = "Price of Fish", y = "Type of Fish") +
  scale_x_continuous(trans = "log10", label = scales::label_number(suffix = "g"), n.breaks = 8) +
  guides(colour = "none") +
  ggtitle("Fish Type Against Price Correlation in Stardew Valley") +
  scale_colour_viridis_d(option = "C") +
  theme_minimal() +
  theme(
        plot.title = element_text(hjust = 0.5, size= 11),
        axis.title = element_text(size = 10))
```

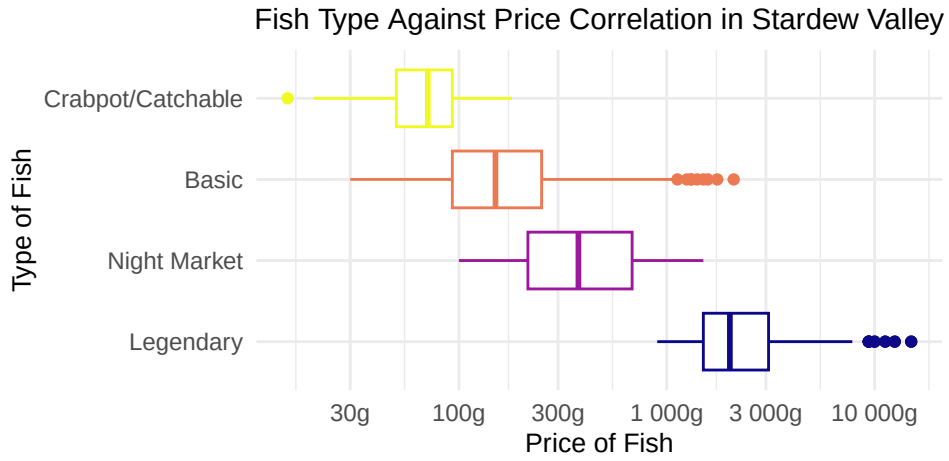

Figure 1: Fish Type Against Price Correlation in Stardew Valley

Figure 1 indicates that legendary fish in Stardew Valley clearly sell the highest, and crabpot/other catchables sell lowest. This is expected, as most catchable items were more common plant items like Algae. Basic and legendary types have more possible outliers that sell for a higher price, which is surprising. There is a steady decrease in price from the highest to lowest column. The night market category has no possible outliers, which makes sense, as there was very little data recorded for this type. Basic fish have a wider range of price, which is also unsurprising as there were a lot of fish in this category compared to the others.

Question 2: What are most fish used for?

```
p2_tibble <- main_tibble %>% # Creating a new tibble counting the frequency of each location
  group_by(usage) %>%
  summarise(
    count = n()
  )

p2 <- p2_tibble %>%
  ggplot(aes(x = fct_reorder(str_to_title(usage), count, .desc = TRUE), y = count,
             fill = fct_reorder(str_to_title(usage), count, .desc = TRUE))) +
  geom_col(width = 0.5) +
  guides(fill = "none") +
  labs(x = "Usage", y = "Count")+
  scale_y_continuous(n.breaks = 7) +
  ggtitle("Density of Fish Usage in Stardew Valley") +
  scale_fill_viridis_d(option = "C") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size=12),
    axis.title = element_text(size=10)
  )
```

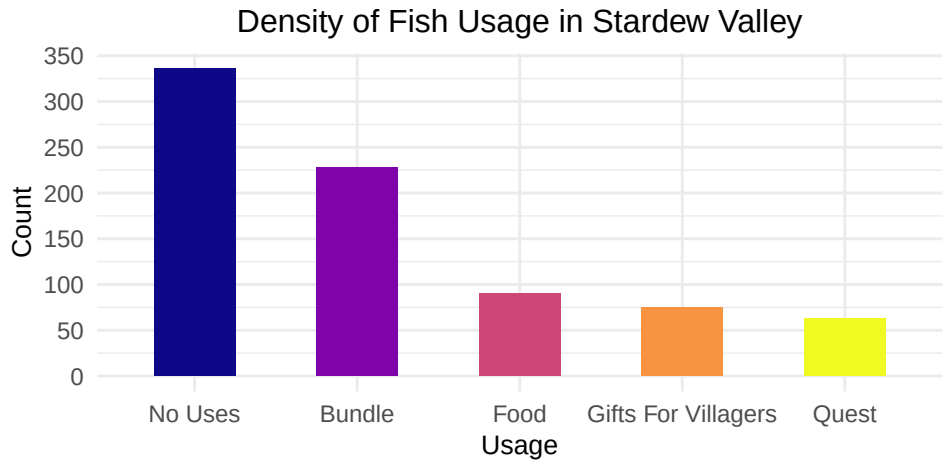

Figure 2: Density of Fish Usage in Stardew Valley

Figure 2 indicates that most fish caught in the game are not used for any specific purpose. After the most frequent column, bundles are the second highest use for caught fish. This could potentially be because the game was designed to encourage players to complete bundles for advantages. There is a noticeable decrease

for the three columns with the lowest frequency- food, gifts for villagers, and quests. Quests are the lowest category that fish are used for in the game.

Question 3: Does fish location correlate to selling price?

The same grouping process for Question 2 was done for the "located" column in the third plot.

```
p3 <- p3_tibble %>%
  ggplot(aes(y = fct_reorder(str_to_title(located), count, .desc = FALSE), x = count,
             fill = fct_reorder(located, count, .desc = FALSE))) +
  geom_col(width = 0.5) +
  guides(fill = "none") +
  labs(x = "Count", y = "Location") +
  scale_x_continuous(breaks = c(0, 25, 50, 75, 100, 125, 150, 175, 200)) +
  ggtitle("Density of Fish Location in Stardew Valley") +
  scale_fill_viridis_d(option = "C") +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, size=12),
    axis.title = element_text(size=10)
  )
```

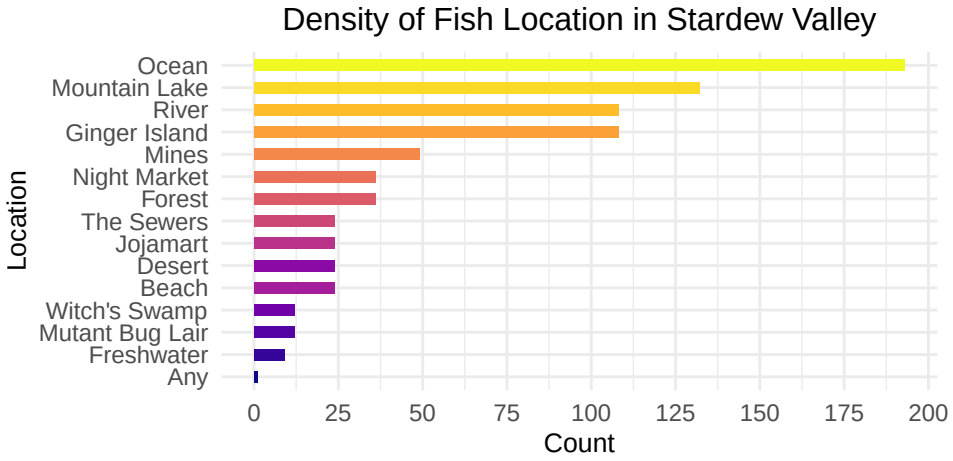

Figure 3: Density of Fish Location in Stardew Valley

Figure 3 indicates that the highest location where fish are caught in Stardew Valley is the ocean. This is expected as it is a larger area within the town. There is a lot of variation between the locations shown in the plot. The plot also suggests that Mountain Lake, River, and Ginger Island are quite close in frequency of fish location. Mutant Bug Lair, Freshwater, and Any (specifying the fish could be found in any location) were locations with the lowest frequency.

Conclusion

I was able to use the generated results to answer my previous questions at the start of the analysis:

1. Do fish sell higher or lower depending on type?
   The boxplot created shows that fish type does seem to impact their selling price. Legendary and night market fish sell for higher, while basic and crabpot/catchable fish tend to sell for lower. There is a steady increase in price between the columns. A player might want to consider looking for legendary fish to catch if they were looking to make more Gold. The results of this plot match my hypothesis for this question.

2. What are most fish used for?
   Results from the previous plot show that most fish in Stardew Valley are not used for any specific reason, and are mainly just collectibles. Furthermore, fish were used frequently for completing bundles, and were less frequently used to craft food, given to villagers, and used to complete quests. There was a distinct increase in frequency for the no uses and bundles category, while the food, gifts for villagers, and quest categories were similar in low frequency. The results of this plot do not support my hypothesis for this question, which was that the most frequent use of fish would be for completing bundles.

3. Where are most fish located among the game map?
   The bar chart used for this question implies that most fish were located in the ocean. Fish that could be found in any location were the least frequent. Depending on where the player is on the map, they could focus on a specific location depending on this data to maximize the fish they catch. The results from this plot support my hypothesis for this question.

The overall results of this analysis show that the player of Stardew Valley can gain both money and a high quantity of fish by focusing on fish type, usage, and location. Type and purpose of fish are clearly tailored to a specific purpose and location. In a game that thrives on economy and earning Gold, the player can use this data to their advantage when fishing in Stardew Valley.

I believe this analysis could be improved if the data was recorded more consistently. Many columns needed to be recoded because of spelling errors or lack of information. In addition, potentially automating the recoding of columns might make the analysis cleaner and easier. For example, I needed to use 17 lines of code when recoding the location column for the fish_detail tibble. It would be very beneficial if there was an easier way to do this.

References

- **Stardew Valley FULL Catalog on Kaggle**
- **Cooking - Stardew Valley Wiki**