Homework #2

Nathaniel Coulter

2025-02-25

Data Frame of Casein and Horsebean

```
library(tidyquant)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
## -- Attaching core tidyquant packages ------------------------ tidyquant 1.0.10 --
## v PerformanceAnalytics 2.0.8      v TTR                0.24.4
## v quantmod             0.4.26     v xts                0.14.1
## -- Conflicts ----------------------------------------- tidyquant_conflicts() --
## x zoo::as.Date()                 masks base::as.Date()
## x zoo::as.Date.numeric()         masks base::as.Date.numeric()
## x PerformanceAnalytics::legend() masks graphics::legend()
## x quantmod::summary()            masks base::summary()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(dplyr)
```

```
##
## ########################## Warning from 'xts' package ##########################
## #                                                                              #
## # The dplyr lag() function breaks how base R's lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won't work correctly.                             #
## #                                                                              #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop           #
## # dplyr from breaking base R's lag() function.                                 #
## #                                                                              #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning.  #
## #                                                                              #
## ################################################################################
##
## Attaching package: 'dplyr'
##
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data("chickwts")

filtered_chicks <- chickwts %>%
  filter(feed %in% c("casein"))

head(filtered_chicks)
```

```
##   weight   feed
## 1    368 casein
## 2    390 casein
## 3    379 casein
## 4    260 casein
## 5    404 casein
## 6    318 casein
```

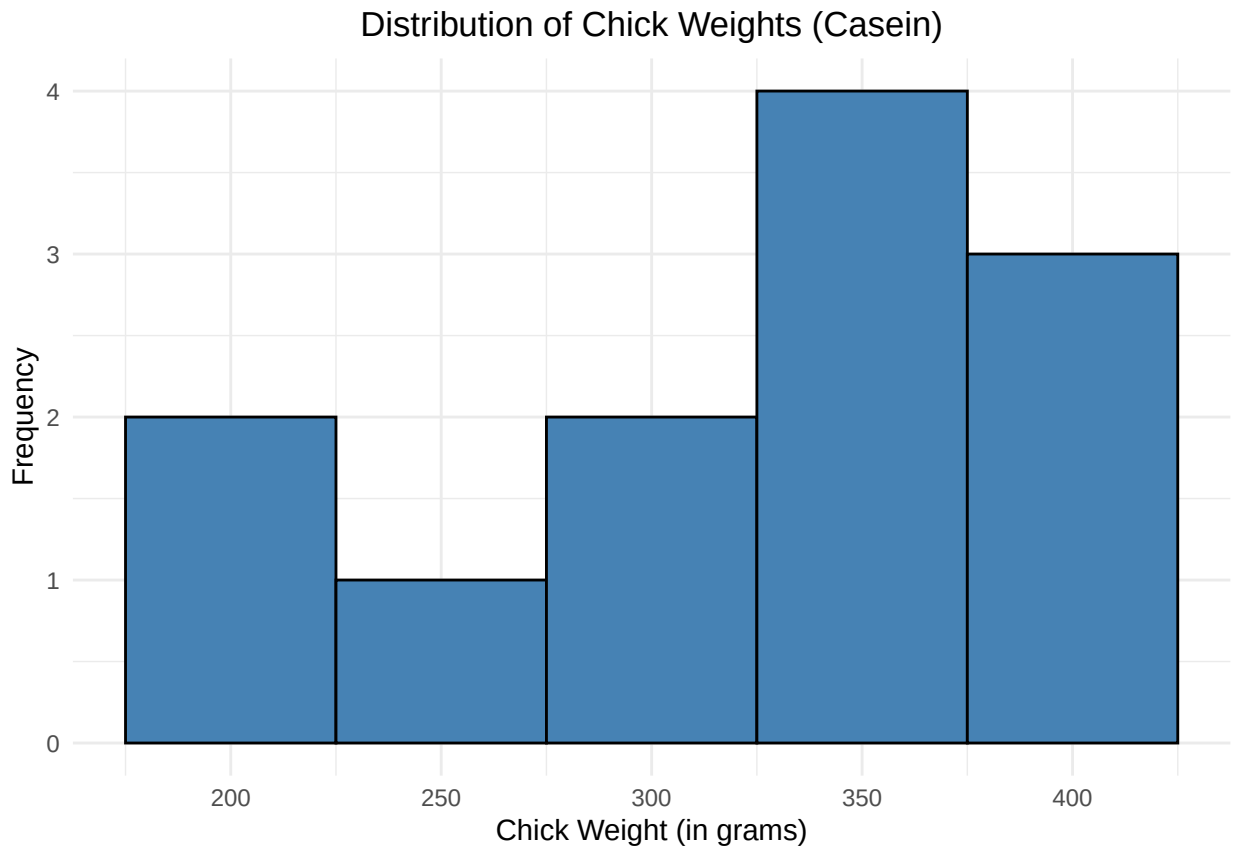
Histogram of chick weights in previous Data Frame

```
library(tidyquant)
library(ggplot2)
library(dplyr)

data("chickwts")

filtered_chicks <- chickwts %>%
  filter(feed %in% c("casein"))

head(filtered_chicks)
```

```
##   weight   feed
## 1    368 casein
## 2    390 casein
## 3    379 casein
## 4    260 casein
## 5    404 casein
## 6    318 casein
```

```
# From Above

#Now adding a plot:
```

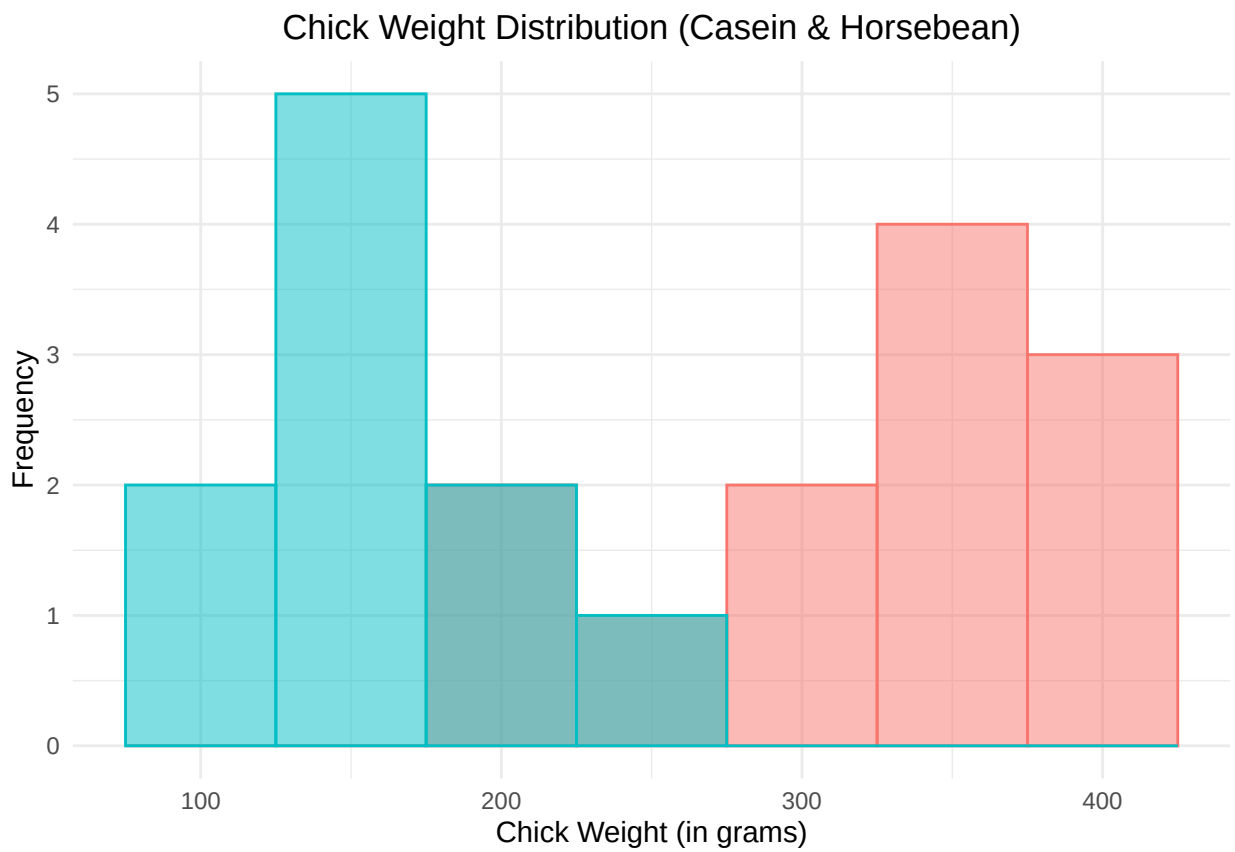
```r
ggplot(filtered_chicks, aes(x = weight)) +
  geom_histogram(binwidth = 50, fill = "steelblue", color = "black") +
  labs(
    title = "Distribution of Chick Weights (Casein)",
    x = "Chick Weight (in grams)",
    y = "Frequency"
  ) +
  theme_minimal()+
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```



Overlaying Histograms

```r
library(dplyr)
library(ggplot2)

data("chickwts")

filtered_chicks <- chickwts %>%
  filter(feed %in% c("casein", "horsebean")) #Using both this time
```

```r
ggplot(filtered_chicks, aes(x = weight, fill = feed, color = feed)) +
  geom_histogram(binwidth = 50, alpha = 0.5, position = "identity") +
  labs(
    title = "Chick Weight Distribution (Casein & Horsebean)",
    x = "Chick Weight (in grams)",
    y = "Frequency",
    fill = "Feed Type",
    color = "Feed Type"
  ) +
  theme_minimal()+
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```



Question 2: (Part A - Data Frame)

```r
# In R Studio on my Data frame it's only showing me AAL.
#I think it's just showing me based on the dates and AAL is first alphabetically.
#For that reason I tried having R fetch and bind each ticker independently.
#But it's still showing me the same...

# So I used the "unique(airline_stocks$symbol)" to verify that each ticker...
```

```r
#...symbol was being retrieved from TidyQuant regardless of what it looks like in R Studio.

library(tidyquant)
library(dplyr)
library(ggplot2)

airline_symbols <- c("AAL", "DAL", "LUV", "UAL")

airline_stocks <- bind_rows(
  tq_get("AAL", get = "stock.prices", from = "2020-01-01", to = "2025-01-31"),
  tq_get("DAL", get = "stock.prices", from = "2020-01-01", to = "2025-01-31"),
  tq_get("LUV", get = "stock.prices", from = "2020-01-01", to = "2025-01-31"),
  tq_get("UAL", get = "stock.prices", from = "2020-01-01", to = "2025-01-31")
)

head(airline_stocks)
```

```
## # A tibble: 6 x 8
##   symbol date        open  high   low close   volume adjusted
##   <chr>  <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 AAL    2020-01-02  29.0  29.3  28.6  29.1  6451100     29.0
## 2 AAL    2020-01-03  28.3  28.3  27.3  27.6 14008900     27.5
## 3 AAL    2020-01-06  27.2  27.5  27.1  27.3  6105800     27.2
## 4 AAL    2020-01-07  27.6  27.7  27.1  27.2  6105900     27.1
## 5 AAL    2020-01-08  27.1  28.1  27.1  27.8 10496800     27.7
## 6 AAL    2020-01-09  28.1  28.2  27.7  28.0  6898900     27.8
```

```r
unique(airline_stocks$symbol)
```

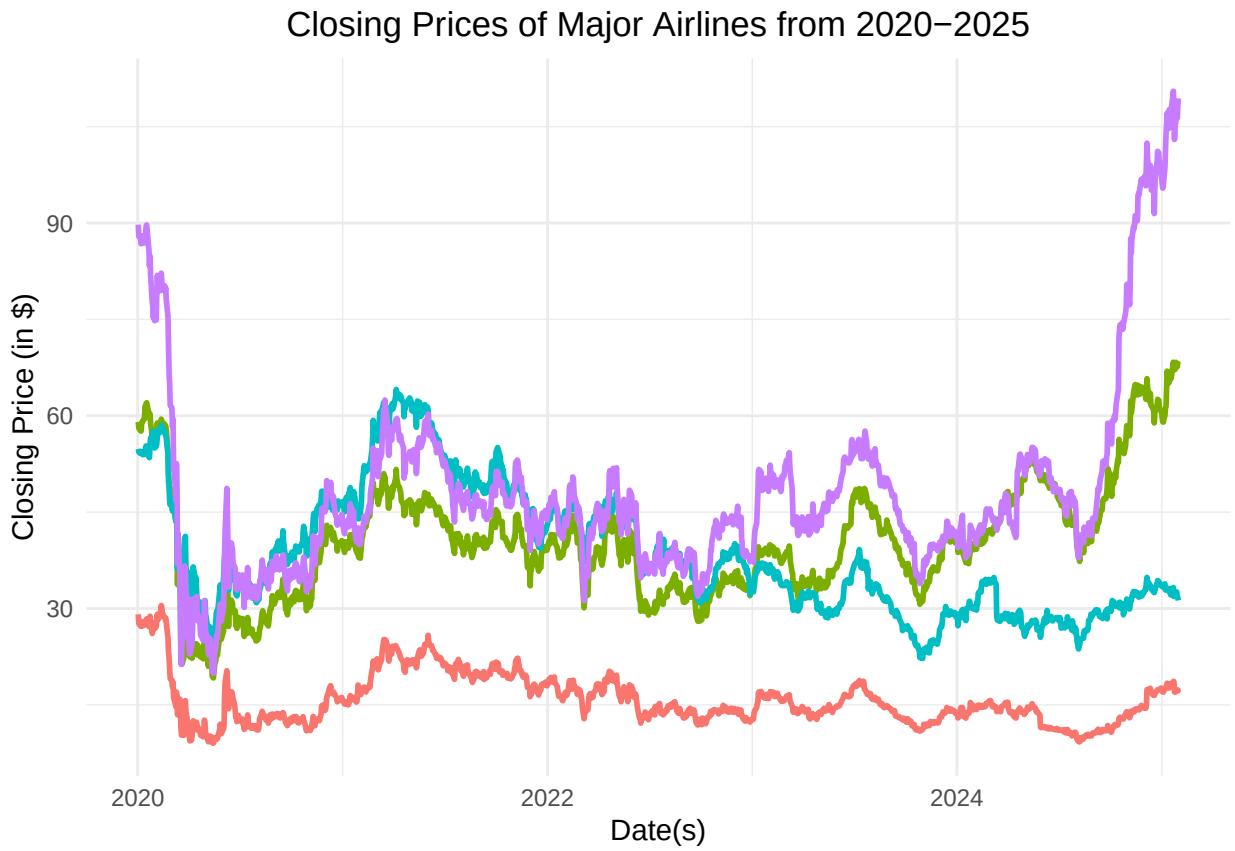
```
## [1] "AAL" "DAL" "LUV" "UAL"
```

Question 2: (Part B - Linegraph)

```r
library(tidyquant)
library(dplyr)
library(ggplot2)

airline_symbols <- c("AAL", "DAL", "LUV", "UAL")

airline_stocks <- tq_get(airline_symbols, get = "stock.prices", from = "2020-01-01", to = "2025-01-31")

ggplot(airline_stocks, aes(x = date, y = close, color = symbol)) +
  geom_line(size = 1) +
  labs(
    title = "Closing Prices of Major Airlines from 2020-2025",
    x = "Date(s)",
    y = "Closing Price (in $)",
    color = "Airline"
  ) +
  theme_minimal()+
```

```
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5)
  )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Question 2: (Part C - Graph only DAL from Data Frame)

```
library(tidyquant)
library(ggplot2)
library(dplyr)

delta_stocks <- airline_stocks %>%
  filter(symbol == "DAL")

ggplot(delta_stocks, aes(x = date, y = close)) +
  geom_line(color = "cyan", size = 1) + # oooh I like this color :)
```

```
labs(
  title = "DAL Closing Prices: 2020-2025",
  x = "Date",
  y = "Closing Price ($)"
) +
theme_minimal()+
theme(
  legend.position = "none",
  plot.title = element_text(hjust = 0.5)
)
```