Final Project

Amish Mishra

December 13, 2020

Bonus Credit

If you can generate the data set into the data structure such as mpg, diamonds or flights as we used for this course (Not covered in the lecture but you can learn how to do this from textbook R for Data Science) and generate all of your results using ggplot function. You will get 20% bonus of your grade for your final project. *Please provide R code before the first question.*

```
# install.packages("readxl")
library(tidyverse)
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.0 --
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## Warning: package 'ggplot2' was built under R version 4.0.3
## -- Conflicts ------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.0.3
```

```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.0.3
```

```
library(TSA)
```

```
## Warning: package 'TSA' was built under R version 4.0.3
##
## Attaching package: 'TSA'
## The following object is masked from 'package:readr':
##
##     spec
## The following objects are masked from 'package:stats':
##
##     acf, arima
## The following object is masked from 'package:utils':
##
##     tar
```

```
turnip_df <- (read_excel("./turnip_price.xlsx"))
turnip_df$day <- rep(c("Sun", "Mon", "Tues", "Wed", "Thurs", "Fri", "Sat"),nrow(turnip_df)/7)
turnip_df$day_idx <- rep(seq(0,6),nrow(turnip_df)/7) # Index of Sunday is 0, Monday is 1, etc.
turnip_df <- turnip_df[, c(1,5,6,2,3,4)] # Reorder columns
head(turnip_df) # Show portion of new data frame
```

```
## # A tibble: 6 x 6
##   date                day   day_idx buy_price sell_price_morni~ sell_price_afte~
##   <dttm>              <chr>   <int>     <dbl>             <dbl>             <dbl>
## 1 2020-04-26 00:00:00 Sun         0       110                NA                NA
## 2 2020-04-27 00:00:00 Mon         1        NA                96                92
## 3 2020-04-28 00:00:00 Tues        2        NA               138               220
## 4 2020-04-29 00:00:00 Wed         3        NA               385               201
## 5 2020-04-30 00:00:00 Thurs       4        NA               123                90
## 6 2020-05-01 00:00:00 Fri         5        NA                NA                NA
```

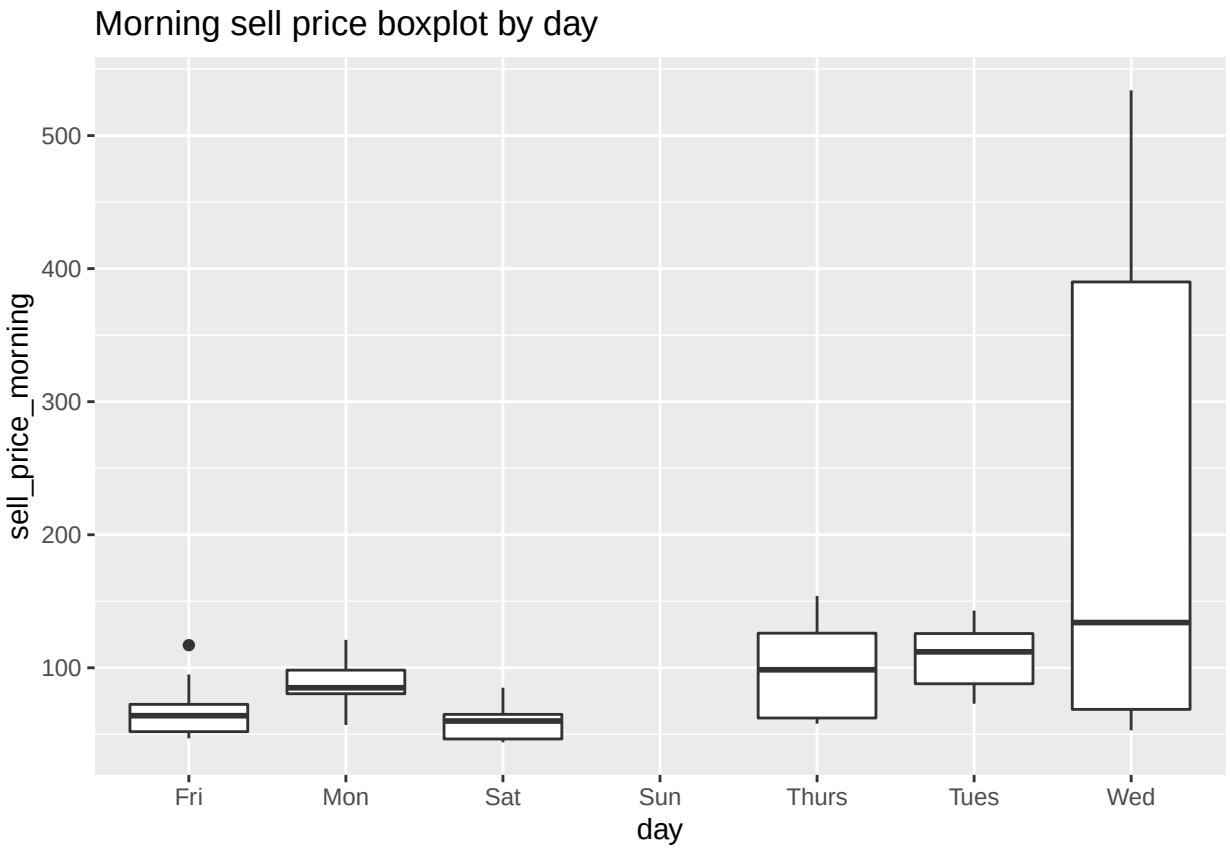
Questions

1. Please find a good way to visualize this data set in order to present better description of this data set using figures.

The method we will use is data transformation to add columns to our data frame such as `day`. Then, we use the visualization techniques of `ggplot` to visualize the data by boxplots and scatter-plots. This is an appropriate way to answer this question because grouping and graphing the data by day makes it much easier to see patterns across multiple weeks.

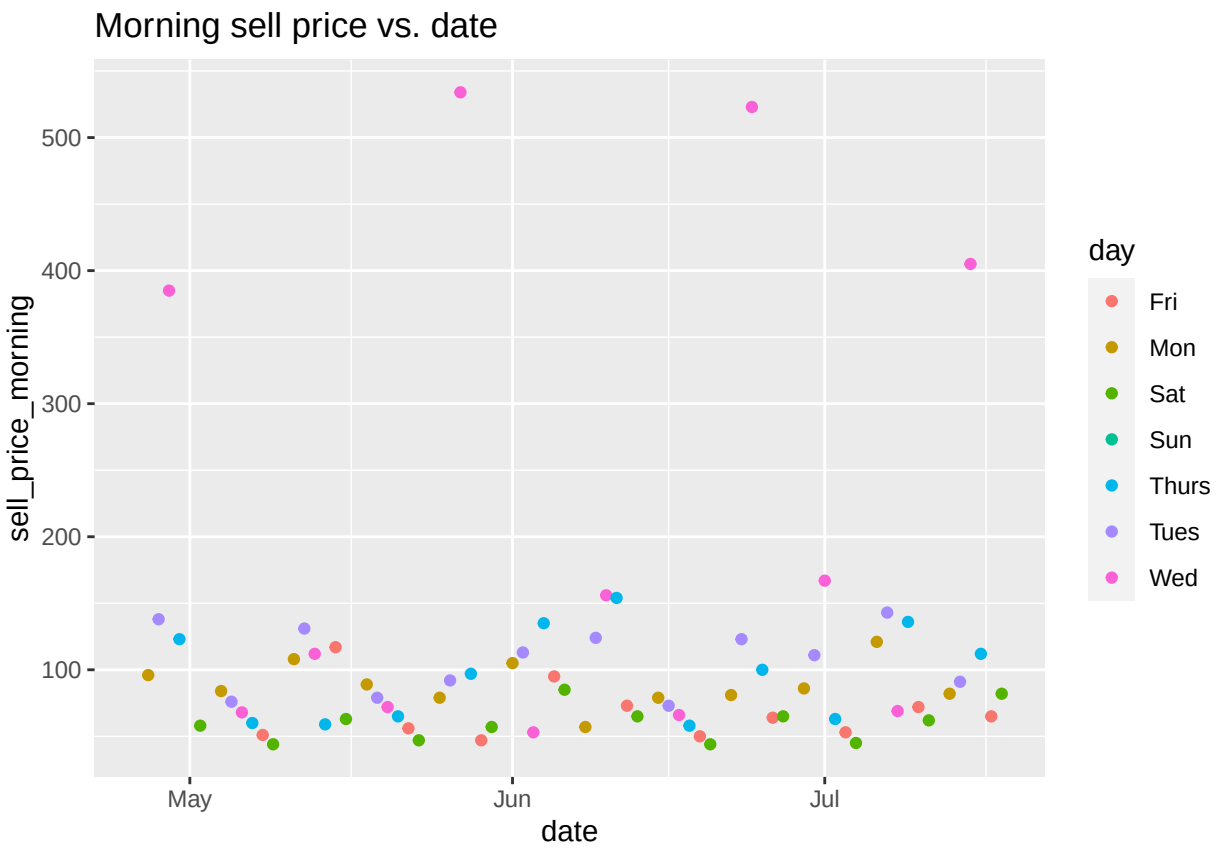
```
# Plots
ggplot(turnip_df, aes(x=day, y=sell_price_morning))+
  geom_boxplot()+
  ggtitle("Morning sell price boxplot by day")
```

```
## Warning: Removed 13 rows containing non-finite values (stat_boxplot).
```



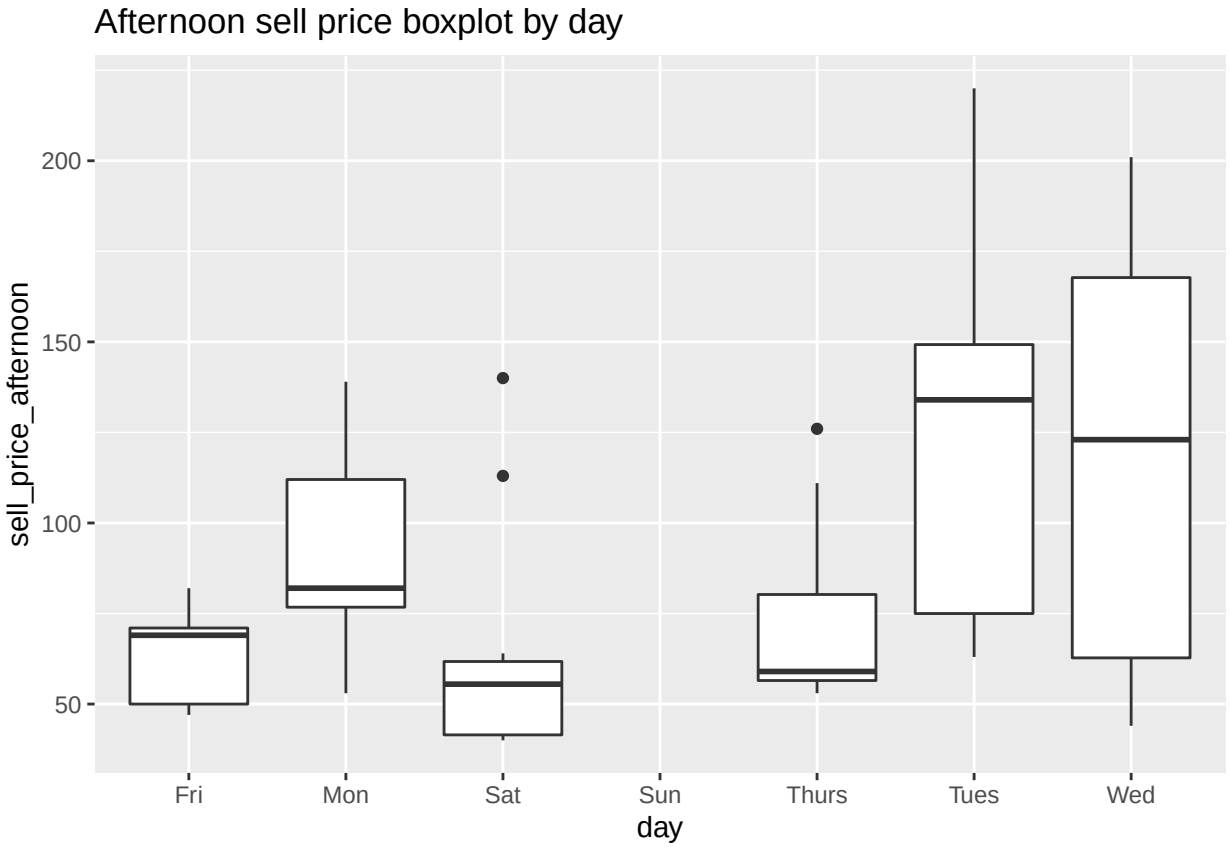

```
ggplot(turnip_df, aes(x=date, y=sell_price_morning, color = day))+
  geom_point()+
  ggtitle("Morning sell price vs. date")
```

```
## Warning: Removed 13 rows containing missing values (geom_point).
```



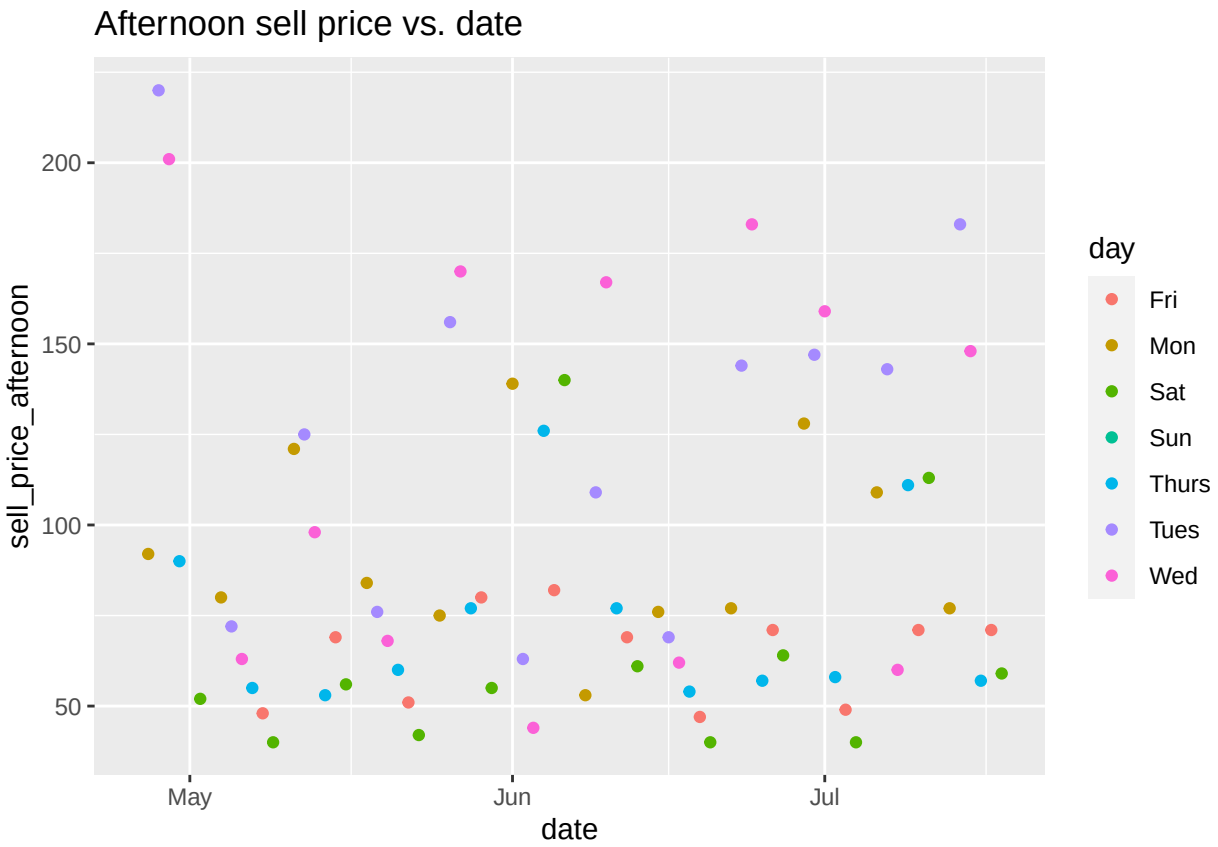

```
ggplot(turnip_df, aes(x=day, y=sell_price_afternoon))+
  geom_boxplot()+
  ggtitle("Afternoon sell price boxplot by day")
```

```
## Warning: Removed 13 rows containing non-finite values (stat_boxplot).
```




```
ggplot(turnip_df, aes(x=date, y=sell_price_afternoon, color = day))+
  geom_point()+
  ggtitle("Afternoon sell price vs. date")
```

```
## Warning: Removed 13 rows containing missing values (geom_point).
```



We see the plots above show some interesting features about the data. For example, the `Morning sell price vs date` graph clearly shows that Wednesdays were high selling price days for the afternoon. Although we would have to do further analysis, from the visual we conclude that the morning selling price of turnips on Wednesdays is generally higher than other days of the week for the mornings.

2. Based on all buying price from the data set, can you estimate mean and the range of the white turnip buying price?

We answer this question by piping the data frame to the `summarise` function, which is able to compute the mean of a given column of the data set. In this case, we have the mean and range of the `buy_price` column with the missing values ignored from the calculation. This is the most appropriate method because the question asks for an estimate of the mean and range of a column of data and the `summarise` function has tools built into it to do specifically that. We remove missing values because most of the missing values are there because those rows are not Sundays (the day when turnips can be bought).

```
turnip_df %>% summarise(mean(buy_price, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   `mean(buy_price, na.rm = TRUE)`
##                             <dbl>
## 1                            98.9
```

```
turnip_df %>% summarise(range(buy_price, na.rm = TRUE))
```

```
## # A tibble: 2 x 1
##   `range(buy_price, na.rm = TRUE)`
##                              <dbl>
## 1                               90
## 2                              110
```

We find that the mean of the `buy_price` column is `98.9` and the range is from `90` to `110`. This means that the average buying price of turnip over the time period the data was collected was about `98.9` with the lowest price being `90` and the highest being `110`.

3. Based on all selling price from the data set, can you estimate mean and the range of the white turnip selling price?

First, we do some data transformations to make a new data frame `combined_sell_price` that has all of the selling prices (morning and afternoon) stacked into one column. We also track which price was taken from the morning and which from the afternoon in a second column. We answer this question by piping the data frame to the `summarise` function, which is able to compute the mean of a given column of the data set. In this case, we have the mean and range of the `sell_price` column with the missing values ignored from the calculation. This is the most appropriate method because the question asks for an estimate of the mean and range of a column of data and the `summarise` function has tools built into it to do specifically that. We remove missing values because most of the missing values are there because those rows are Sundays (not the day when turnips can be sold).

```
# Make one column of all selling prices
temp_df <- turnip_df %>%
  select(sell_price_morning) %>%
  bind_rows(
    turnip_df %>%
      transmute(sell_price_morning = sell_price_afternoon)
  )
combined_sell_price = rename(temp_df, sell_price = sell_price_morning)
combined_sell_price$morn_aft <- rep(c("morning","afternoon"), each = 84)
head(combined_sell_price)
```

```
## # A tibble: 6 x 2
##   sell_price morn_aft
##        <dbl> <chr>
## 1         NA morning
## 2         96 morning
## 3        138 morning
## 4        385 morning
## 5        123 morning
## 6         NA morning
```

```
combined_sell_price %>% summarise(mean(sell_price, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   `mean(sell_price, na.rm = TRUE)`
##                              <dbl>
## 1                             98.4
```

```
combined_sell_price %>% summarise(range(sell_price, na.rm = TRUE))
```

```
## # A tibble: 2 x 1
##   `range(sell_price, na.rm = TRUE)`
##                               <dbl>
## 1                                40
## 2                               534
```

We find that the mean of the `sell_price` column is `98.4` and the range is from `40` to `534`. This means that the average selling price of turnip over the time period the data was collected was about `98.4` with the lowest price being `40` and the highest being `534`.

4. Will the white turnip selling price in the afternoon be significantly higher than the price in the morning?

We first justify why we choose to use a 2-sample t-test rather than a paired t-test. There is no a priori reason to assume that the prices in the afternoon of a given day are affected by the price in the morning. So, we assume that the two columns are independent. Now, we perform a 2-sample t-test with the null hypothesis that the mean of the selling price in the morning (μ_m) and the mean of the selling price in the afternoon (μ_a) are the same. The alternative hypothesis will be that the mean of the selling price in the afternoon is higher than the mean of the selling price in the morning. Our significance level will be `0.05`. These are summarized in the equations below:

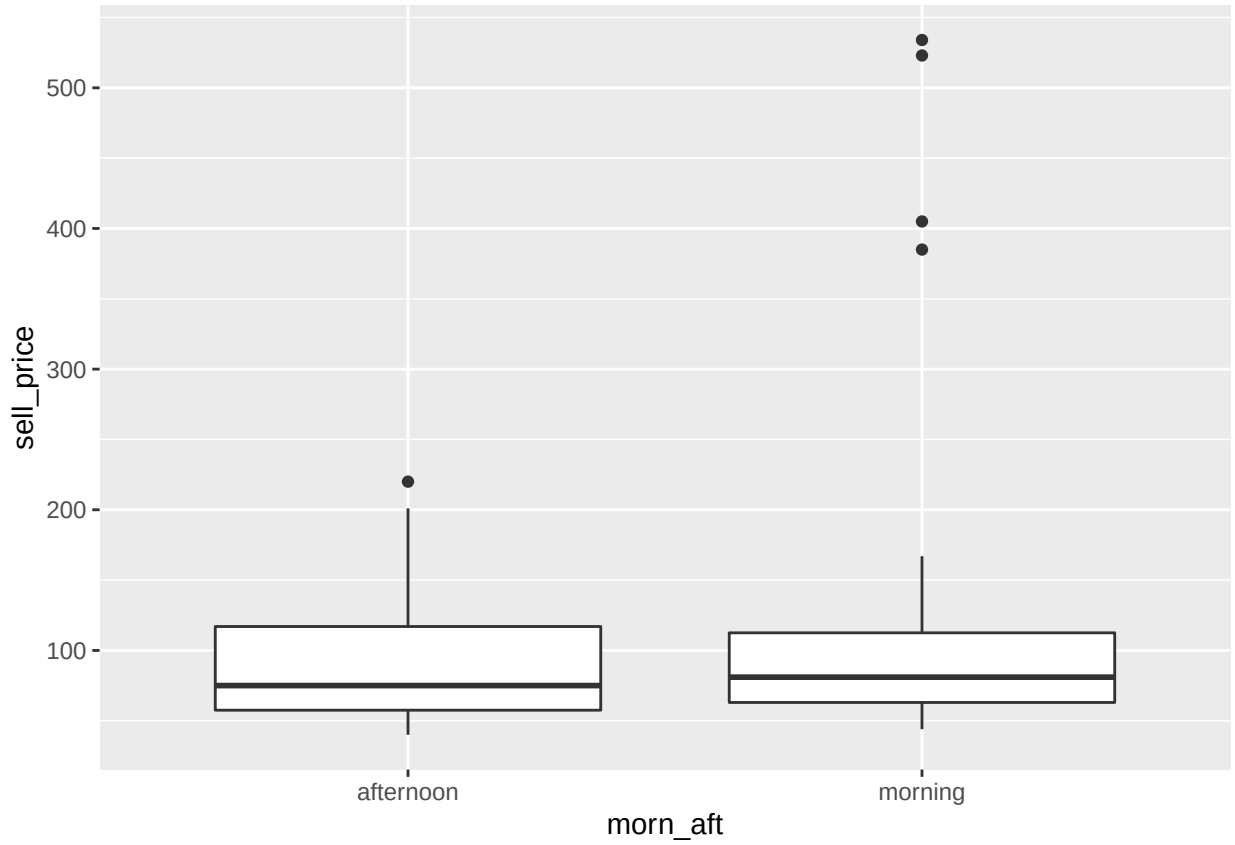
$$H_0 : \mu_a = \mu_m$$

$$H_a : \mu_a > \mu_m.$$

We check the normality and equal variance assumptions of the t-test below.

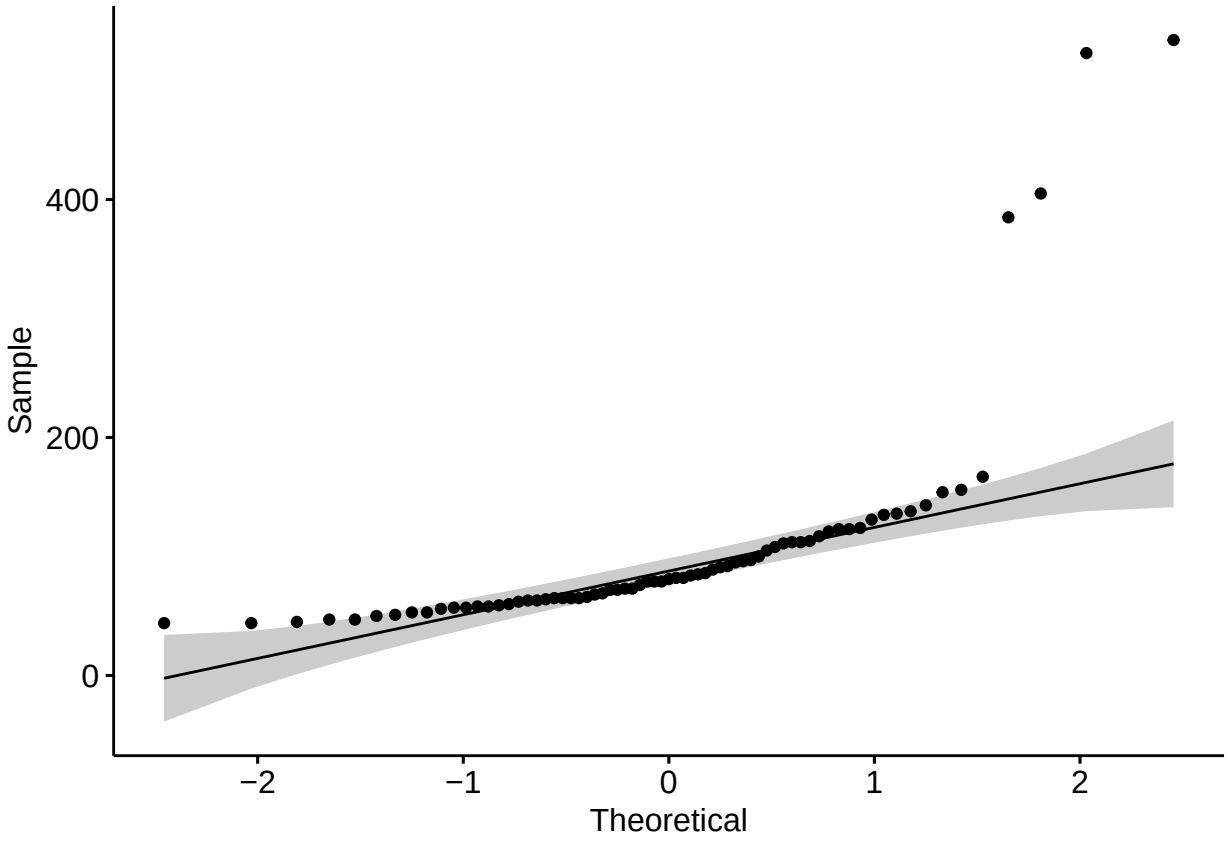
```
ggplot(combined_sell_price, aes(x=morn_aft, y=sell_price)) +
  geom_boxplot()
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```




```
ggqqplot(turnip_df$sell_price_morning)
```

```
## Warning: Removed 13 rows containing non-finite values (stat_qq).
## Warning: Removed 13 rows containing non-finite values (stat_qq_line).
## Warning: Removed 13 rows containing non-finite values (stat_qq_line).
```

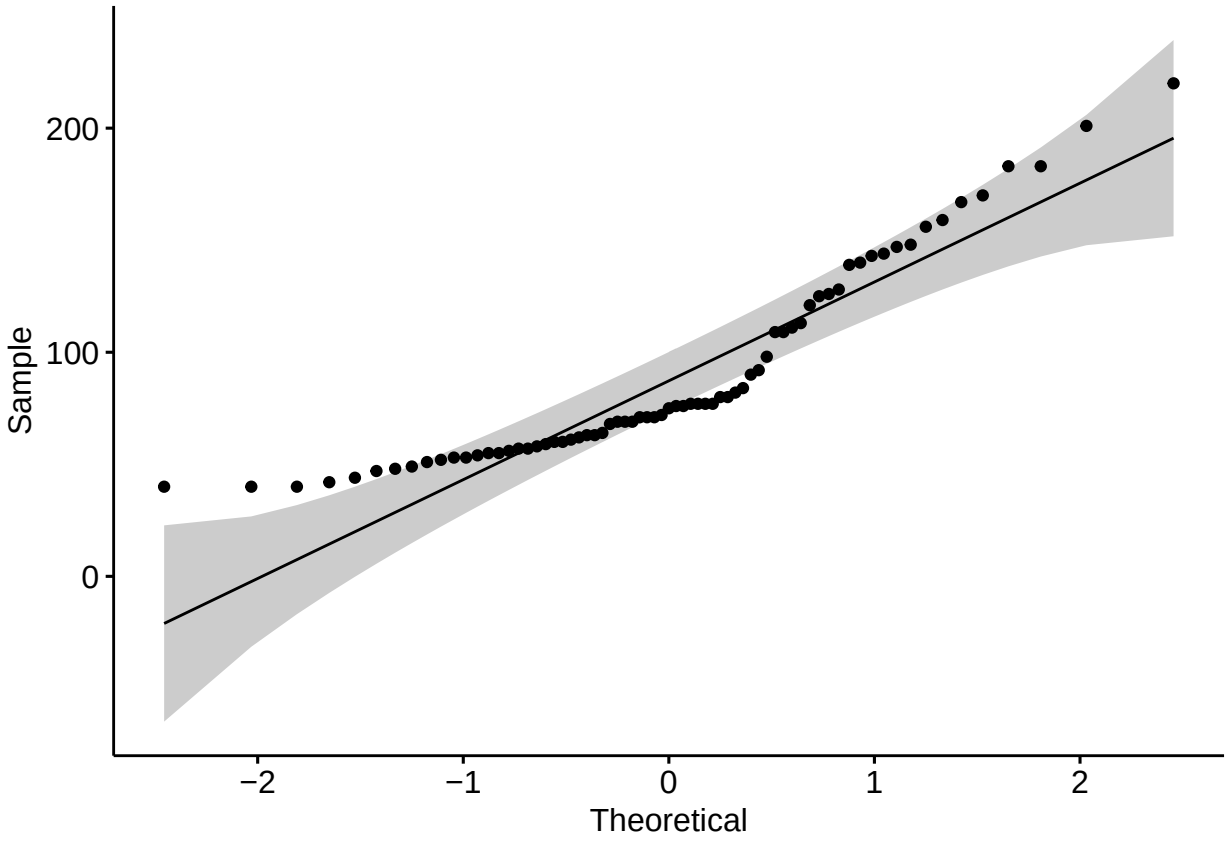



```
ggqqplot(turnip_df$sell_price_afternoon)
```

```
## Warning: Removed 13 rows containing non-finite values (stat_qq).
```

```
## Warning: Removed 13 rows containing non-finite values (stat_qq_line).
```

```
## Warning: Removed 13 rows containing non-finite values (stat_qq_line).
```



We find from the boxplot that there are many outliers in the dataset, but the vertical thickness of the boxes suggests that the variance of the two data columns are similar. With the exception of a handful of points from the afternoon price normality plot, both normal probability plots remain close to linearity. This suggests that the distributions of the both columns of data are decently normal. Hence, we will move forward with the t-test with caution. We will also perform the Wilcoxon Mann-Whitney test which is not affected by the outliers and does not assume the normality of the input data. This is also an appropriate test because we have 2 samples and do not wish to assume much about the nature of the distribution they are taken from.

```
t.test(data = combined_sell_price, sell_price ~ morn_aft, var.equal = TRUE, alternative = "less")
```

```
##
##  Two Sample t-test
##
## data:  sell_price by morn_aft
## t = -1.384, df = 140, p-value = 0.08427
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##      -Inf 3.338121
## sample estimates:
## mean in group afternoon   mean in group morning
##                89.94366                106.94366
```

```
wilcox.test(data = combined_sell_price, sell_price ~ morn_aft, var.equal = TRUE,conf.int = TRUE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  sell_price by morn_aft
```

```
## W = 2314.5, p-value = 0.4017
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -14.999963   6.000041
## sample estimates:
## difference in location
##              -4.000016
```

We find the results that the p-value of the t-test came out as `0.084` and the p-value of the Wilcoxon Mann-Whitney came out as `0.40`. Since the p-values in both tests came out to be larger than `0.05` (our significance level), we conclude that we fail to reject the null hypothesis (H_0). Hence, the sample of data we have does not provide sufficient evidence to conclude that the selling price of white turnips in the afternoon will be significantly greater than that of the turnips in the morning.

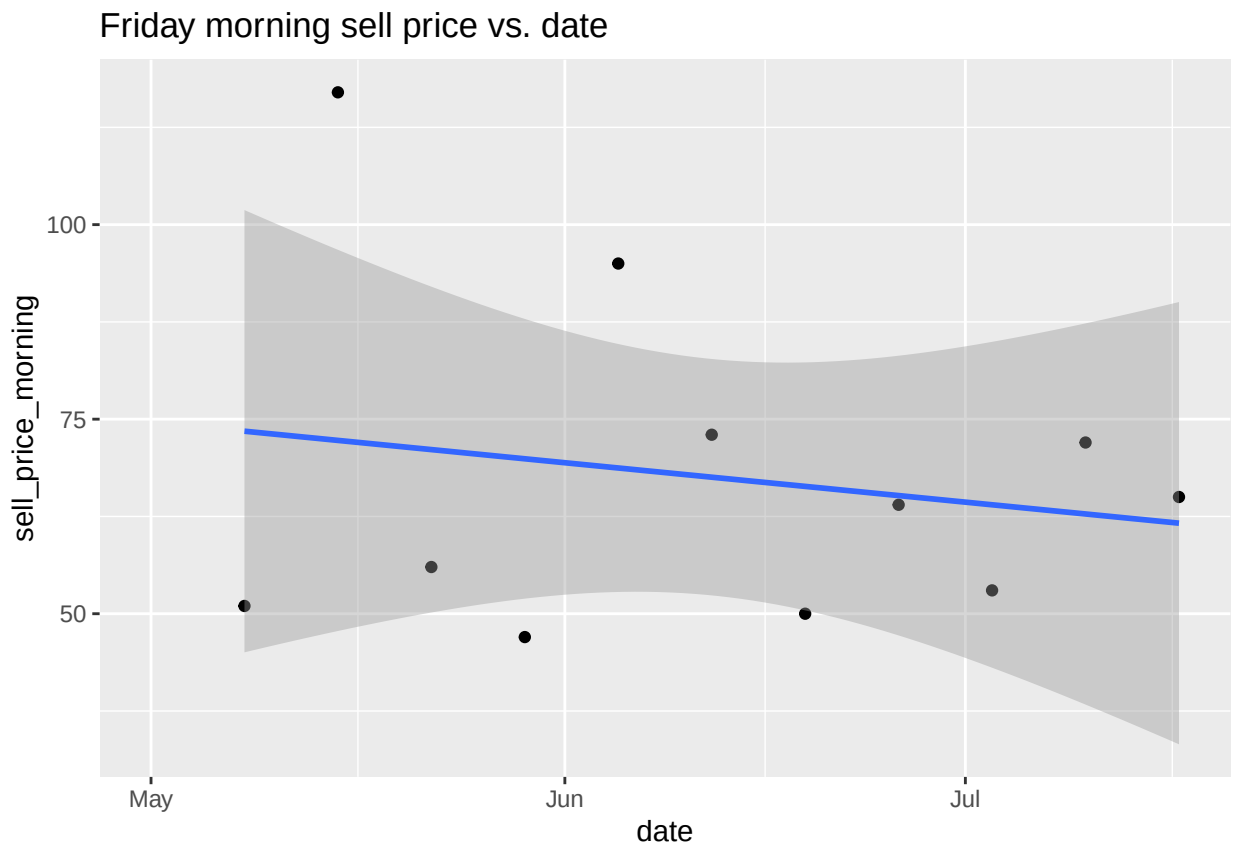
5. The white turnip selling prices on 5/1 is missing, this is because the stalk-market will be closed for upgrading one day if the player play this game and open the market for 30 days. So selling price for upgrading is unavailable for one day. Can you find a way to impute reasonable number for the missing selling prices for morning and afternoon?

We perform some Exploratory Data Analysis (EDA) to see if we can inform a reasonable estimate by the data itself. This is an appropriate method because we are trying to impute information from the data, so we are required to begin with exploring the data itself. We start by first noting that 5/1 is a Friday. Now, we can plot all of the Friday turnip selling prices on a graph and look for patterns.

```
friday_df <- filter(turnip_df, day == "Fri")
ggplot(friday_df, aes(x=date, y=sell_price_morning))+
  geom_point()+
   geom_smooth(method='lm', formula= y~x)+
  ggtitle("Friday morning sell price vs. date")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

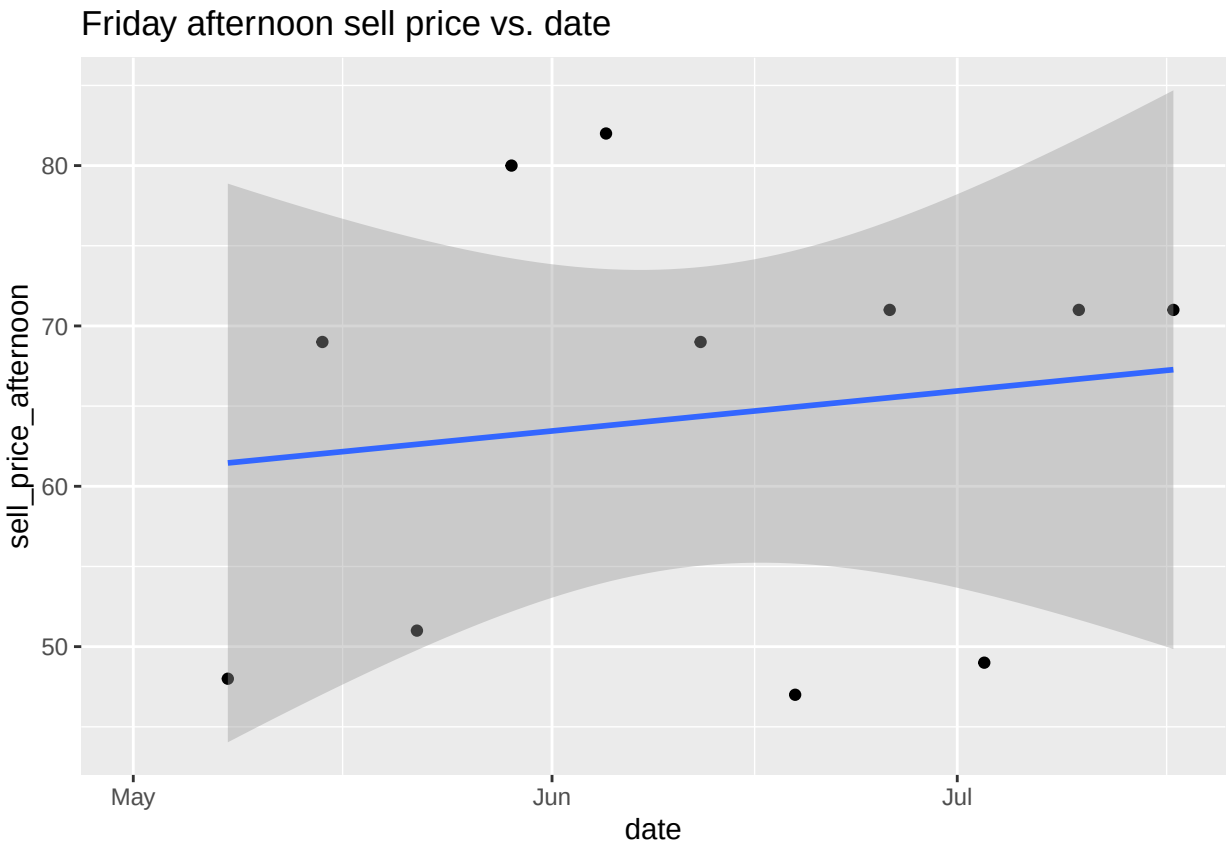
```
## Warning: Removed 1 rows containing missing values (geom_point).
```




```
ggplot(friday_df, aes(x=date, y=sell_price_afternoon))+
  geom_point()+
   geom_smooth(method='lm', formula= y~x)+
  ggtitle("Friday afternoon sell price vs. date")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Unfortunately, the plots do not show any sort of significant trend. Hence, we decide that an accessible and reasonable way to impute a number for the morning and the afternoon selling price for 5/1 is to place the average of the morning prices for the morning sell price of 5/1 and similarly for the afternoon. This is reasonable because we do not know presumptuously if the prices around a given day affect the prices on that day. Hence, we will proceed with caution and use the conservative method of imputing the morning/afternoon selling prices from the means of their respective data sets.

```
turnip_df %>% summarise(mean(sell_price_morning, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   `mean(sell_price_morning, na.rm = TRUE)`
##                                      <dbl>
## 1                                     107.
```

```
turnip_df %>% summarise(mean(sell_price_afternoon, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   `mean(sell_price_afternoon, na.rm = TRUE)`
##                                        <dbl>
## 1                                       89.9
```

We find that the mean of the morning selling prices of the turnips is `106.9` and the mean of the afternoon selling prices is `89.9`. Hence, we suggest that the morning selling price of turnips on 5/1 ought to be `106.9` and in the afternoon be `89.9`.

6. From the game rule of introduction, we know that white turnips must be sold within a week, otherwise they will rot by the following Sunday. So that means the buying price will only affected by the selling price for the following 6 days. Based on the data set we have, what is the probability the player will earn money for buying the white turnip in this game? Please describe your way in detail to answer this

question and showing your result using table or figure to support your finding.

We begin by first doing some data transformations to extract some needed information. We calculate a column of profit for selling the turnips in the morning of a given day of a given week and a column for the afternoon profits. This is made by subtracting the selling price for the morning/afternoon from the buying price at the beginning of the week.

```
temp_df <- turnip_df
sun_df <- filter(temp_df, day == "Sun")
buy_price_vec <- pull(sun_df, buy_price)
temp_df$buy_price <- rep(buy_price_vec, each = 7)

temp_df <- temp_df %>%
  mutate(
    profit_morning = sell_price_morning - buy_price,
    profit_afternoon = sell_price_afternoon - buy_price
  )
turnip_df$profit_morning <- temp_df$profit_morning
turnip_df$profit_afternoon <- temp_df$profit_afternoon
head(turnip_df)
```

```
## # A tibble: 6 x 8
##   date                day   day_idx buy_price sell_price_morn~ sell_price_afte~
##   <dttm>              <chr>   <int>     <dbl>            <dbl>             <dbl>
## 1 2020-04-26 00:00:00 Sun         0       110               NA                NA
## 2 2020-04-27 00:00:00 Mon         1        NA               96                92
## 3 2020-04-28 00:00:00 Tues        2        NA              138               220
## 4 2020-04-29 00:00:00 Wed         3        NA              385               201
## 5 2020-04-30 00:00:00 Thurs       4        NA              123                90
## 6 2020-05-01 00:00:00 Fri         5        NA               NA                NA
## # ... with 2 more variables: profit_morning <dbl>, profit_afternoon <dbl>
```

Now, the method we use is by constructing a table of how many profits were positive overall and how many were negative overall. By overall, we mean the number of positive profits total from the morning and the afternoon profit columns. This method is appropriate because it captures the number of positive profit sales, which can be used to find the proportion of positive profits. This would give us a statistic of what chance a user has of making a positive profit on a given sale playing this game. In the table, the `-1` column is a negative profit (a loss of money) and `1` is a positive profit column (made money on the sale).

```
neg_pos_morning <- table(sign(turnip_df$profit_morning))
neg_pos_afternoon <- table(sign(turnip_df$profit_afternoon))
neg_pos_sum <- neg_pos_morning + neg_pos_afternoon
neg_pos_sum
```

```
##
## -1  1
## 93 49
```

```
(percent_prop <- neg_pos_sum[2]/(neg_pos_sum[1]+neg_pos_sum[2]))
```

```
##         1
## 0.3450704
```

Lastly, we extract the proportion of positive profits from the total number of times a sale was made (sum of the number of morning and evening profits). This proportion comes out at `0.345`. So, we have found that there is a 34.5% chance of making a profit playing this game, based on the data provided.

7. If we call the buying price on Sunday and all the following selling prices from Monday through Saturday

a period, Is there any specific pattern for selling price for all periods? If there exist patterns, then how many? (If you google the white turnip for this game, it is not hard to find the answer, and I can tell you there are four patterns for selling price, but you have to use your own way to find patterns).

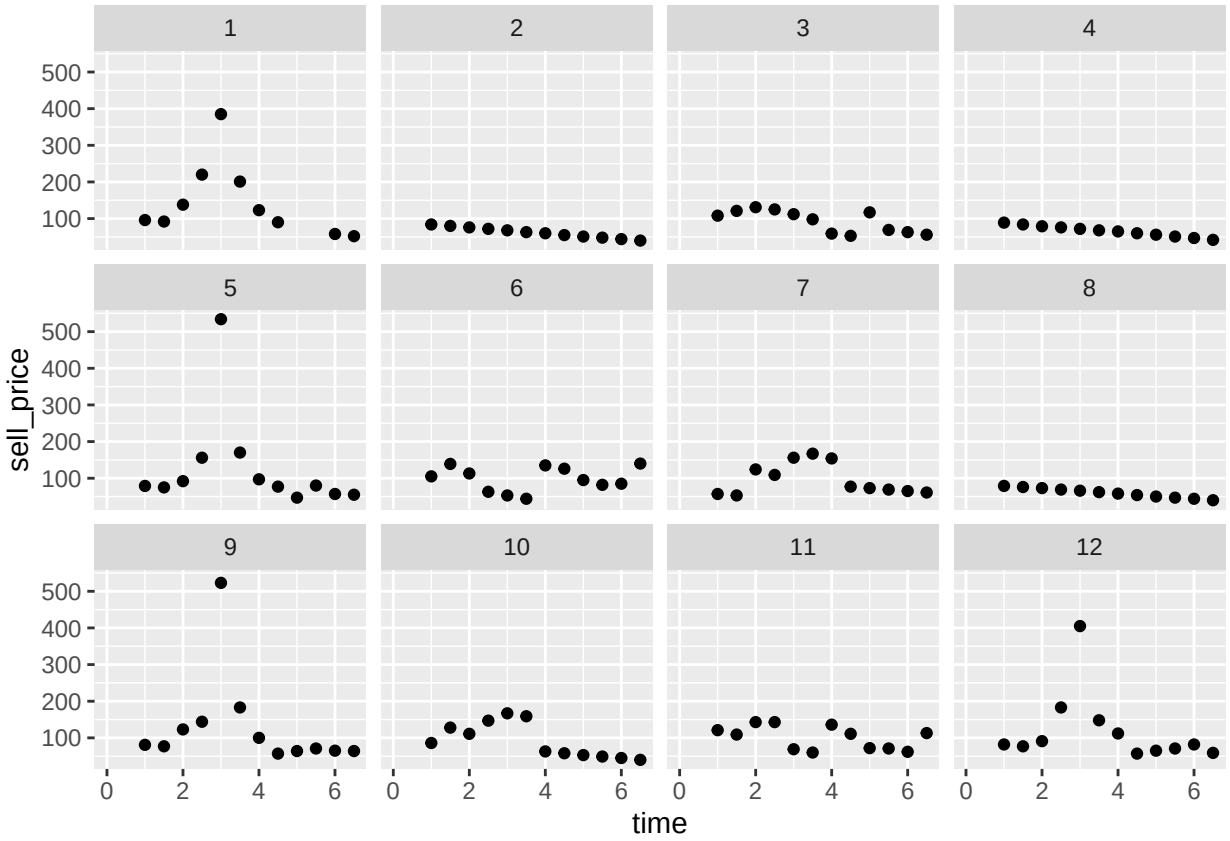
We first begin by constructing a new data frame named `flat_sell_price` which captures the selling prices all into one column and then another column captures the time of day the price was put. An integer time corresponds to the selling price being from the morning column and a half-integer time corresponds to the selling price being from the afternoon column. Another column named `period` captures the 12 weeks the data was taken. Our method is to use `ggplot` with `facet_wrap` to visualize the data. This is appropriate because we wish to see the selling price vary as y-variable on a plot against the time during the week. Further, we wish to compare the different periods (weeks) with one another to compare any pattern in selling prices that we may find.

```
flat_sell_price <- data.frame(sell_price = as.vector(t(select(turnip_df, sell_price_morning, sell_price
flat_sell_price$period <- rep(seq(1,12), each = 14)
flat_sell_price$time <- rep(seq(0,6.5,0.5), 12)
head(flat_sell_price)
```

```
##   sell_price period time
## 1         NA      1  0.0
## 2         NA      1  0.5
## 3         96      1  1.0
## 4         92      1  1.5
## 5        138      1  2.0
## 6        220      1  2.5
```

```
ggplot(data = flat_sell_price, mapping = aes(x = time, y = sell_price))+
  geom_point()+
  facet_wrap(~ period)
```

```
## Warning: Removed 26 rows containing missing values (geom_point).
```



From the following plots, we see that there are 4 patterns we can identify across the 12 weeks.

- Spike: demonstrated in periods 1,5,9, and 12
- Small Spike: demonstrated in periods 10 and 7
- Steady Decrease: demonstrated in periods 2, 4, and 8
- Random: demonstrated in periods 6, 3, and 11.

We define "Small Spikes" as periods where the selling price has general increase and then a decrease to it. Period 11 would not qualify as such since the points rise up, then down, then up, then down, and then up towards the end. We describe this sort of period as "Random". "Spike" and "Stead Decrease" should be visually clear from the output.

8. Now we know there are four different patterns for selling prices every week, will the specific pattern for this week affect the probability of certain patterns for the following week? For example, if this week happens with pattern number 1, what is the probability of the other three patterns next week. Please answer this question by using the results (patterns) from previous question and use as underlying patterns.

We make a data frame to track which pattern is followed by what other pattern. In the following data frame, for a given column, the row represents how many times the row name pattern came after the column name pattern. For example, in the "Decrease" column, the 2 represents the fact that a "Decrease" pattern was followed by a "Spike" pattern 2 times in the given data set periods.

```
(patterns_df <- data.frame(Spike = c(0,1,1,1),
                           Small_Spike = c(0,0,1,1),
                           Decrease = c(2, 0, 0, 1),
                           Random = c(1, 1, 1, 0),
                           row.names = c("Spike", "Small_Spike", "Decrease", "Random")))
```

```
##              Spike Small_Spike Decrease Random
## Spike            0           0        2      1
## Small_Spike      1           0        0      1
## Decrease         1           1        0      1
## Random           1           1        1      0
```

From this data, now we calculate the probabilities by dividing each column by the sum of the numbers of that column. This method is appropriate because this gives us the proportion of times pattern X was followed by pattern Y out of the total number of patterns that followed pattern X.

```
pattern_prob <- patterns_df
pattern_prob <- patterns_df %>%
  mutate(Spike = Spike/sum(Spike),
         Small_Spike = Small_Spike/sum(Small_Spike),
         Decrease = Decrease/sum(Decrease),
         Random = Random/sum(Random))
rownames(pattern_prob) = c("Spike", "Small_Spike", "Decrease", "Random")
pattern_prob
```

```
##                 Spike Small_Spike  Decrease    Random
## Spike       0.0000000         0.0 0.6666667 0.3333333
## Small_Spike 0.3333333         0.0 0.0000000 0.3333333
## Decrease    0.3333333         0.5 0.0000000 0.3333333
## Random      0.3333333         0.5 0.3333333 0.0000000
```

Now, we have a probability table. To demonstrate how to read it, let's take for example that the current week exhibited a "Decrease" pattern in its turnip selling prices. We guess that the following week has a 2 in 3 chance of being a "Spike" and a 1 in 3 chance of being a "Random". To summarize, an entry of the table represents the proportion/chance the column-name pattern will be followed by the row-name pattern.

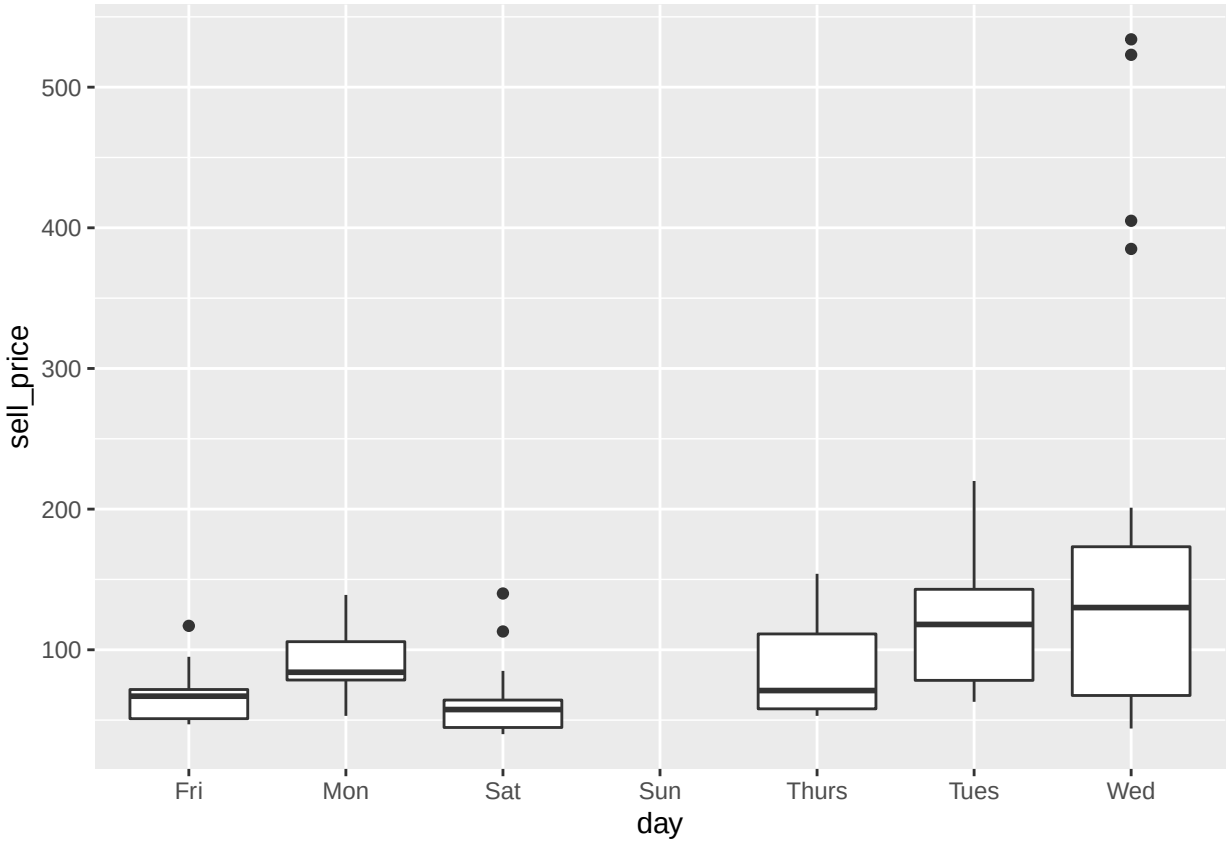
9. Will certain days for selling price from Monday to Saturday are tend to be higher or lower than all the other days? (For example, the selling price for all Monday from the data set are significantly lower than the selling price in other days, or the selling price for all Thursday are significantly higher than the selling price in other days). Please investigate the hypothesis and show your finding.

We first do a box plot of the selling prices grouped by the day of the week they occurred. We also run a non-parametric Kruskal-Wallis test because we don't have a guarantee that the samples for each day are independent of one another. Hence, a non-parametric test is appropriate and advisable. Further, we are comparing more than 2 groups' means. Our null hypothesis is that there is no difference in the mean selling price across the 6 days Monday through Saturday data we have and the alternative hypothesis is that there is a difference. Our significance level is 0.05. We write μ_i for the i^{th} day of the week where $i = 1$ is Monday.

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_6$$

```
flat_sell_price$day <- rep(c("Sun", "Mon", "Tues", "Wed", "Thurs", "Fri", "Sat"),
             times = 12,
             each = 2)
ggplot(flat_sell_price, mapping = aes(x = day, y = sell_price))+
  geom_boxplot()
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```




```
kruskal.test(sell_price ~ day, data = flat_sell_price)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  sell_price by day
## Kruskal-Wallis chi-squared = 47.301, df = 5, p-value = 4.933e-09
```

We can rest assured that Sunday did not skew our test because all of the Sunday sell price rows were `NA`, so they were not accounted for in the test. This is confirmed by the degrees of freedom of the rank-sum test since it should have 1 less degree of freedom than the number of groups. In our case, for 6 valid groups we ought to have 5 degrees of freedom, which is what the test reports above in the output.

After running the test, we find that the p-value of 4.933×10^{-9} is far less than the significance level `0.05`, which means that we reject the null hypothesis and conclude that there is a difference in the average selling price of the turnips across varying days of the week. From the box plot, the largest median and outliers appear in the Wednesday group. This suggests that the selling price on Wednesday tend to be higher than the other days. Furthermore, from the boxplot we see that Saturdays tend to be bad days to sell because it has the lowest median selling price.

10. If the goal of this project is to predict the selling price in order to earn a lot of money or not losing any money in this game, can you predict the selling prices from Thursday to Saturday if we already have the buying price on Sunday and all selling prices from Monday to Wednesday? Try to predict the price using two following independent scenarios:

For this problem, we use a time series analysis approach. We first make a time series by using the methods from this article[1]. The time series is named `sell_price_time_series`. The time series method is appropriate

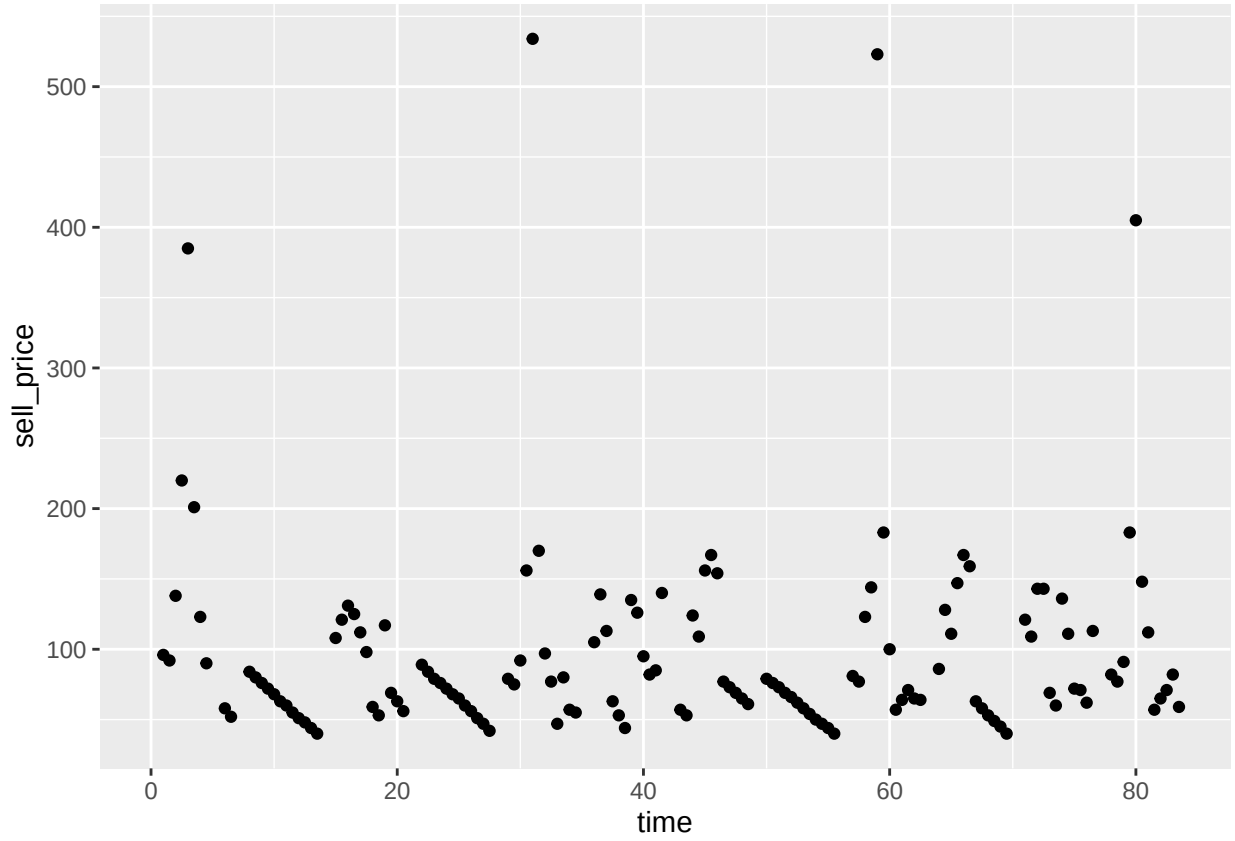
[1] https://a-little-book-of-r-for-time-series.readthedocs.io/en/latest/src/timeseries.html

for this problem because the turnip selling prices occur at different values as time moves forward. Since the event of selling turnips repeats every week, we make the assumption that the prices will exhibit some sort of periodic behavior. We divide each week into 14 points representing Sunday morning, Sunday afternoon, Monday morning, Monday afternoon, etc. After this, we fit a harmonic function to our time series and analyze the output.

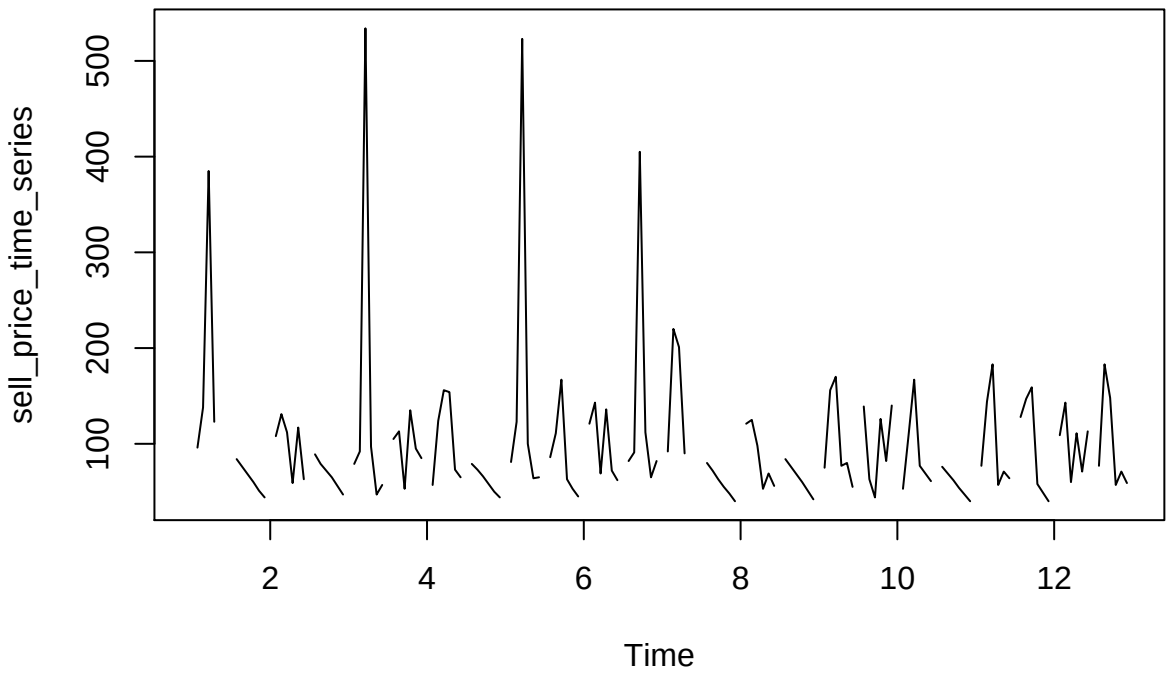
```
sell_price_over_time <- flat_sell_price
sell_price_over_time$time <-seq(0,83.5,0.5)

ggplot(data = sell_price_over_time, mapping = aes(x = time, y = sell_price))+
  geom_point()
```

```
## Warning: Removed 26 rows containing missing values (geom_point).
```




```
sell_price_time_series <- ts(combined_sell_price$sell_price, frequency = 14)
plot.ts(sell_price_time_series)
```

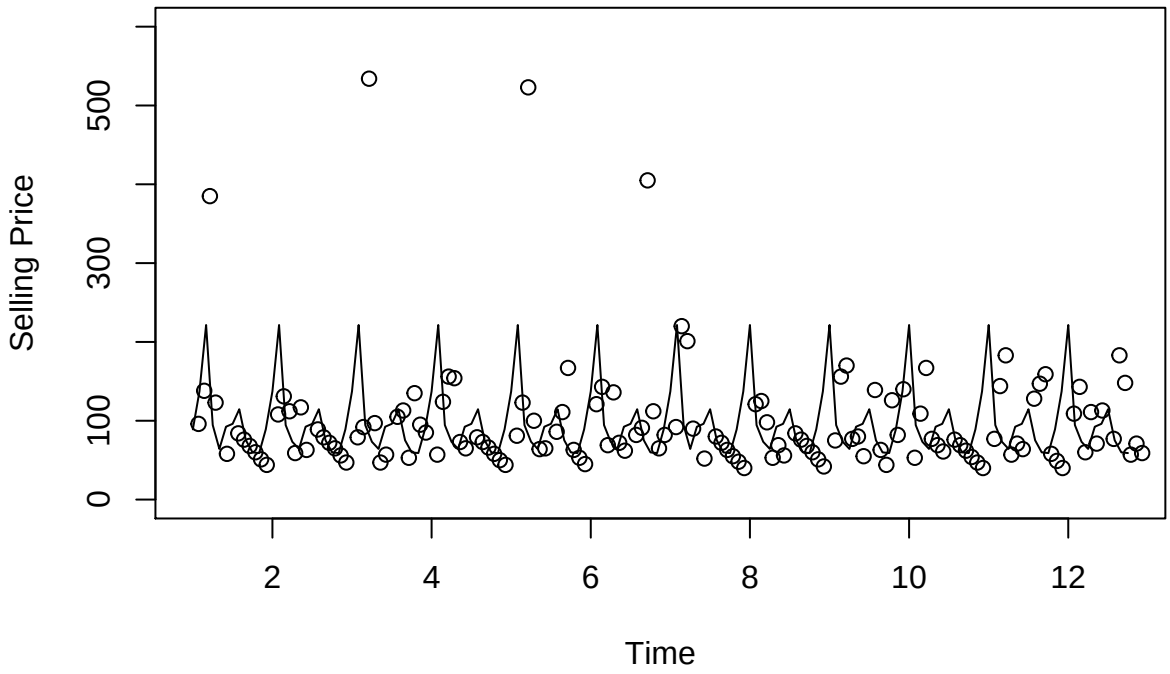



```
har. = harmonic(sell_price_time_series,6)
model = lm(sell_price_time_series ~ har.)
summary(model)
```

```
##
## Call:
## lm(formula = sell_price_time_series ~ har.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -161.500  -18.812   -8.625    6.863  312.500
##
## Coefficients: (1 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        95.4087     5.8168  16.402  < 2e-16 ***
## har.cos(2*pi*t)    11.4853    11.6337   0.987 0.325357
## har.cos(4*pi*t)   -26.9583     9.5409  -2.826 0.005465 **
## har.cos(6*pi*t)    -4.3287    11.8242  -0.366 0.714892
## har.cos(8*pi*t)    10.8439     9.6487   1.124 0.263142
## har.cos(10*pi*t)   18.4935    11.7083   1.580 0.116647
## har.cos(12*pi*t)        NA         NA      NA       NA
## har.sin(2*pi*t)    23.8020     6.7705   3.516 0.000605 ***
## har.sin(4*pi*t)    30.7829     6.8027   4.525 1.35e-05 ***
## har.sin(6*pi*t)   -10.4331     6.7302  -1.550 0.123527
## har.sin(8*pi*t)   -16.6809     6.7302  -2.479 0.014474 *
## har.sin(10*pi*t)    0.4136     6.8027   0.061 0.951610
```

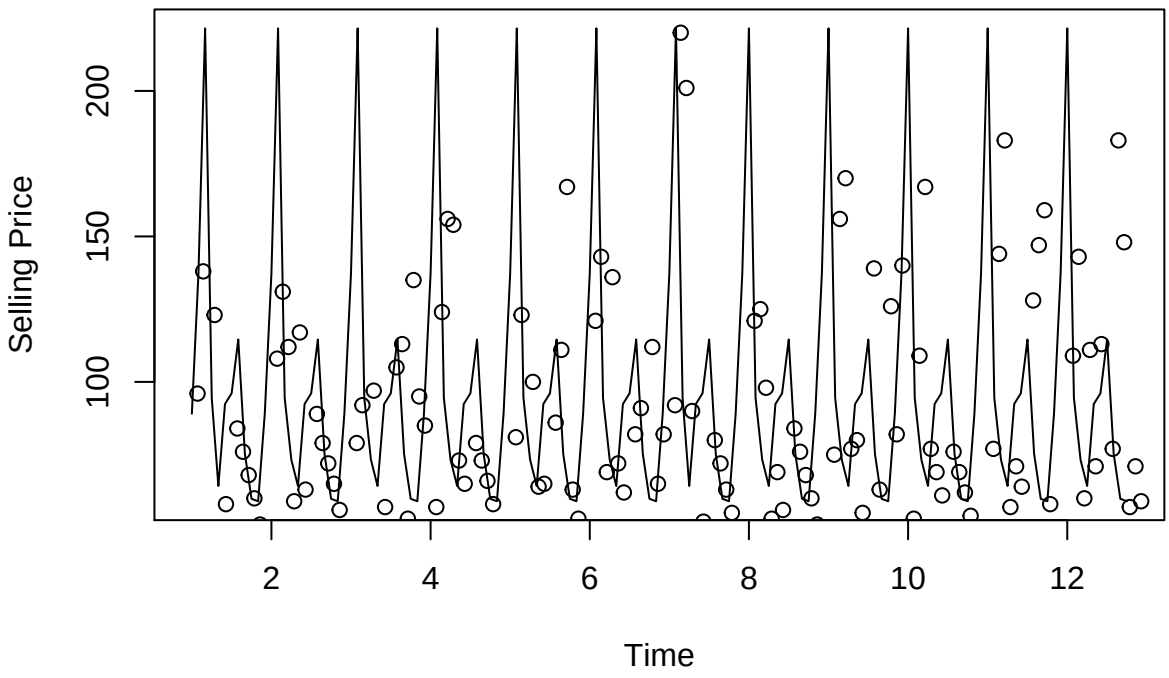
```
## har.sin(12*pi*t)  15.5760     6.7705   2.301 0.023009 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.52 on 130 degrees of freedom
##   (26 observations deleted due to missingness)
## Multiple R-squared:  0.3527, Adjusted R-squared:  0.298
## F-statistic:  6.44 on 11 and 130 DF,  p-value: 1.821e-08
```

```
plot(ts(fitted(model),freq = 12),
     ylab = "Selling Price", type = "l", ylim = c(0,600))
points(sell_price_time_series)
```



The first plot outputted is a scatter plot showing the progression of turnip selling prices as time went by. Our analysis regression output shows an R-squared value of `0.35`. This means that our model fits the data set quite poorly. This is reflected in the visual comparison of the plots. The harmonic fit to the data set neglects to capture the outliers into the model. However, the graph does show that the regression fit to the data is quite close to the selling prices below 300. We can see that in a zoomed in graph below:

```
plot(ts(fitted(model),freq = 12),
     ylab = "Selling Price", type = "l")
points(sell_price_time_series)
```



Generally speaking, there are spikes where there ought to be spikes and dips where the data selling prices are low. Regardless, we use this model to make a prediction for the following scenarios.

Note that by the nature of the model we chose, our predictions will not be influenced by the prices on Monday, Tuesday, and Wednesday. Thus, the following two scenarios have the same price predictions because we did not need to use the provided selling price data for the harmonic model we have.

Scenario 1: Buying price is 93 on Sunday, and the selling prices are 140 and 127 on Mondays, 183 and 212 on Tuesday and 158 and 83 on Wednesday?

```
model_prices <- predict(model)
head(model_prices, 15)
```

```
##         2         3         4         5         7         9        10        11
##  89.08333 137.33333 221.50000  94.50000  64.25000  92.41667  96.08333 114.58333
##        12        13        14        16        17        18        19
##  75.25000  59.83333  59.00000  89.08333 137.33333 221.50000  94.50000
```

We see from the values of the model that the series repeats the same sequence `2-14` over and over. This is because `0` and `1` correspond to Sundays and there's no selling price data for that day. Hence, we need to simply look at what numbers correspond with Thursday morning, Thursday afternoon, Friday morning, Friday afternoon, Saturday morning, and Saturday afternoon.

```
newdata = data.frame("day"=rep(c("Thurs", "Fri", "Sat"), each = 2),
                     "time_of_day"=rep(c("morning","afternoon"),3),
                     "time_series_time" = seq(8,13))
newdata %>% mutate(prediction = model_prices[time_series_time-1])
```

```
##     day time_of_day time_series_time prediction
```

```
## 1 Thurs     morning                8   96.08333
## 2 Thurs   afternoon                9  114.58333
## 3   Fri     morning               10   75.25000
## 4   Fri   afternoon               11   59.83333
## 5   Sat     morning               12   59.00000
## 6   Sat   afternoon               13   89.08333
```

Thus, we make the prediction that the prices for the days following Wednesday will be `96, 115, 75, 60, 59, 89`.

Although these predictions are based on a model that made the assumption the prices are periodic generally, we have some reasonable predictions of prices for the following weeks. Based on our predictions, for a buying price of `93`, the player ought to sell the turnips on Thursday morning or afternoon to make positive profit.

Scenario 2: Buying price is 107 on Sunday, and the selling price are 104 and 138 on Mondays, 65 and 58 on Tuesday and 109 and 101 on Wednesday? You can rely on some machine learning algorithm or modeling procedure for prediction of the selling prices.

We use the same time series output model for this scenario as well.

```
model_prices <- predict(model)
head(model_prices, 15)
```

```
##         2         3         4         5         7         9        10        11
##  89.08333 137.33333 221.50000  94.50000  64.25000  92.41667  96.08333 114.58333
##        12        13        14        16        17        18        19
##  75.25000  59.83333  59.00000  89.08333 137.33333 221.50000  94.50000
```

We see from the values of the model that the series repeats the same sequence `2-14` over and over. This is because `0` and `1` correspond to Sundays and there's no selling price data for that day. Hence, we need to simply look at what numbers correspond with Thursday morning, Thursday afternoon, Friday morning, Friday afternoon, Saturday morning, and Saturday afternoon.

```
newdata = data.frame("day"=rep(c("Thurs", "Fri", "Sat"), each = 2),
                     "time_of_day"=rep(c("morning","afternoon"),3),
                     "time_series_time" = seq(8,13))
newdata %>% mutate(prediction = model_prices[time_series_time-1])
```

```
##     day time_of_day time_series_time prediction
## 1 Thurs     morning                8   96.08333
## 2 Thurs   afternoon                9  114.58333
## 3   Fri     morning               10   75.25000
## 4   Fri   afternoon               11   59.83333
## 5   Sat     morning               12   59.00000
## 6   Sat   afternoon               13   89.08333
```

Thus, we make the prediction that the prices for the days following Wednesday will be `96, 115, 75, 60, 59, 89`.

Although these predictions are based on a model that made the assumption the prices are periodic generally, we have some reasonable predictions of prices for the following weeks. Based on our predictions, for a buying price of `107`, the player ought to sell the turnips on Thursday afternoon to make positive profit.