Kabeer Intro Data Project

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.2     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.3     v tibble    3.2.1
v lubridate 1.9.2     v tidyr     1.3.0
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to beco
```

Executive Summary

This report is to increase all inefficiencies of an ice cream shop in St A. Many different statistical methods were used including two sample t-tests, odds ratios, linear models, power tests, and more. The important things to take away from this report is that it is very focused to this specific scenario. The power testing is not reproducible, however it is not meant to be due to the specific conditions of the location of the store. Some of the variables have a much larger coefficient than others and the model created clearly shows the result of this in predictions. Parsing their individual affects combined with other individual affects has not been conducted.

Introduction

A St Andrews ice cream shop has been collecting data each day in 2023 from the number of ice creams and number of hot drinks sold. They would like to be able to make better predictions of these sales in future, in order to improve the efficiency of stock ordering. I have been employed by the ice cream shop to analyse the data and provide a report that answers the questions attached in the appendix.

Methods

Part 2

I first conducted a basic analysis to see how many days the shop went selling less than 200 number of ice creams by counting the number of days below and total days data was collected. I then used the more efficient mean function to more easily display the result in a percentage while keeping the count and summarize so that the true number of days below 200 ice cream sales and days can be seen to provide a much more real display of what is going on with the shop. To predict the future number of days with ice cream sales below 200 I used a binomial test and then a 95% confidence interval. The binomial test is appropriate because there is a binary outcome (over or under 200 ice cream sales), each day is independent, and because I am trying to find the expected proportion. I did this EXACT same thing with total sales for part 2B because it follows the same logic. I manually calculate the odds ratios and confidence interval for part 2C. I did not use epitools because groups 1 and 2 would just be inverses of each otherw. I first summed up the respective groups and got the basic odds for each probability occurring. I treated ice cream being sold as a "win". I found the standard error, took the log, and then exponentiated back to scale to find the confidence intervals. For part 2D, I succinctly showing significant difference without hours of useless tests by taking advantage of logic.

Part 3

I first filter out the data for sales on weekdays vs weekends. I then run a t-test to see whether there is an expected difference in sales from weekdays to weekends. For part 3B; to compute the power of the t-test, I find the standard deviation and sample size for each group, weekend and weekday. I use this to find the pooled standard deviation using the formula $s_p = \sqrt{\frac{(n_1-1)s_1^2+(n_2-1)s_2^2}{n_1+n_2-2}}$

I then calculated delta by subtracting the means from the t-test from part 3A. This was inputted into the power t-test function to find the power of the test. For part 3C I kept the same process above and just kept changing delta by guessing and checking different values to get a 90% power level. Ditto as above for part 3D.

Part 4

To see the relationship with all the factors and ice cream sales, I run a linear model. It gives most the information needed to interpret the effect of each variable and which to dive into. To see how different factors affect the demand for ice cream, I make a data frame for each of the four situations the owner wants information for. This dataframe holds the criteria provided for each situation which is then used to predict a large number of variations of outcomes. This is then averaged to see what the possible outcome could be in an accurate way (law of large numbers). The upper and lower intervals are included in this and help risk manage any predictions.

Results

Part 2A

```
[1] 0.3495146
```

```
probability of success
             0.3495146
```

```
[1] 0.2581794 0.4497513
attr(,"conf.level")
[1] 0.95
```

This shows that the estimated probability of there being a day below 200 ice cream sales is ~34%. Since the p-value is less than the alpha, I reject the null and assume that there is a statistically significant difference. The confidence interval shows that I am 95% confident that the expected proportion of having a day under 200 ice cream sales is between ~26% to ~45%

Part 2B

```
probability of success
              0.184466
```

```
[1] 0.1148697 0.2729811
attr(,"conf.level")
[1] 0.95
```

This shows that the estimated probability of there being a day below 200 total sales is ~18%. Since the p-value is less than the alpha, I reject the null and assume that there is a statistically significant difference. The confidence interval shows that I am 95% confident that the expected proportion of having a day under 200 ice cream sales is between ~11% to ~27%. It makes sense that this interval has both bounds much lower than the ice cream sale test above since the amount of product being counted per day increases greatly. For part 2C: the results were a 95% confidence interval for ice cream to be sold in January from ~0.80 to ~0.93, with a ratio of ~.86. This means that - on average - it is around 14% less likely for the shop to sell an ice cream compared to hot drink for this month (while being 95% certain that that the likelihood is between 7%-20% less). Since 1 is not included, it is presumed to be significant. For August the 95% interval was from ~3.89 to ~4.36, the ratio being 4.11. This means that - on average - it is 4.11 times more likely to sell an ice cream over a hot drink in summer (while being 95%

certain that the likelihood is between 3.89 to 4.36 times selling a hot drink). Part 2D shows there is a significant different in odds. August has both ends of its 95% confidence interval above 1 while January has both ends of its 95% confidence interval below 1. This shows a wild change. This also makes complete sense. Although it is an ice cream shop, it makes sense how hot drinks can slightly outsell ice cream during one of the coldest months of the year while during summer, the amount of ice cream sold dwarfs the amount of hot drinks sold.

Part 3

```
    Welch Two Sample t-test

data:  weekdays_sales and weekends_sales
t = -6.8102, df = 93.836, p-value = 9.171e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -356.0242 -195.2857
sample estimates:
mean of x mean of y
 324.4038  600.0588
```

Conducted a t-test that shows there is a significant difference between average sales on weekends and weekdays. The average sales on weekdays is multiple hundreds lower than on weekends. The test shows that there is 95% confidence that sales on weekdays are between ~195 to ~396 units less than on weekends! For part 3B: the power level of the current test is almost 100%. Although it is impossible to predict something 100% of the times, our model is close to doing so in this situation. It is important to note that the model is not standardized. This implies the test conducted is very reliable. For part 3C, the difference in means had to be changed to ~134 to reach a 90% confidence interval. This change in effect size is hard to understand since the observed difference is 276. In part 3D, n had to be reduced to 13 in order to get a 90% confidence interval. We basically had to make the model much less accurate to reach this level since it is so close to being perfect. The extremely large standard deviation and the extreme consistent difference in both sets of data is why this is occurring.

Part 4

Part 4 results are the most interesting. For part A, although it is warm and May, the number of ice cream predicted (142 coeff) to be demanded is no where near the demand of ice cream on a 28 degree day in April (700 coeff with 470 lower bound and 933 upper bound). The wind also seems to ruin the mood of people wanting ice cream. The demand we predict for a day in Sep that has 35km/h wind has a coeff of just 9, with the lower bound being -175 and the upper bound being 195. This shows that certain temperatures and wind conditions seem affect the amount of ice cream sold more than time of month.

Discussion

A lot has been found out in this project. I think the area of main opportunity is capitalizing on good variable days. When the weather and time of week conditions line up to be warm, the store sells ice cream at an unimaginable rate. When the weather is cold, it is harder to sell but still possible. When it is holiday time, it is shown and predicted that people love to get ice cream and drinks. One possible violation made is not having a reproducible part 2. Although this model is very good at prediction, it is not standardized. The outcomes of this model should only be used here and with caution. One suggestion I have is trying to combine just two or three variables and removing the effects of all others. It may be possible to find a stronger trend line in that form.

Appendix

```r
#laod libraries
library(tidyverse)
library(ggplot2)
library(lubridate)
library(epitools)

set.seed(7)
setwd("/Users/kabeermotwani/Library/Mobile Documents/com~apple~CloudDocs/intro project")

#load data
sales <- read.csv("sales_data.csv")
sales <- sales %>%
  mutate(
    temperature = as.numeric(temperature),
    humidity = as.numeric(humidity),
    windspeed = as.numeric(windspeed),
    month_name = as.factor(month_name),
    weekend = as.factor(weekend),
    bank_holiday = as.factor(bank_holiday),
    school_holidays = as.factor(school_holidays),
    icecream_sales = as.integer(icecream_sales),
    hotdrink_sales = as.integer(hotdrink_sales)
  )
```

```r
head(sales,5)
```

```
  month_name weekend bank_holiday school_holidays temperature windspeed
1        Jan       0            1               1         5.7       5.5
2        Jan       1            0               0         7.3       2.3
3        Jan       0            0               0         5.4      17.7
4        Jan       0            0               0         0.0       7.6
5        Jan       1            0               0         2.3       0.3
  humidity hotdrink_sales icecream_sales
1       20            335            391
2       75            149            201
3       57            102             57
4       22            131             22
5       65            220            180
```

```r
#counts days where less than 200 ice cream sales
icecream200count <- sum(sales$icecream_sales<200)
icecream200count
```

```
[1] 36
```

```r
#counts total days of ice cream sales
total_days_icecream_sales <- sales %>%
  summarise(total_days = n())
total_days_icecream_sales
```

```
  total_days
1        103
```

```r
#calculates average days of sales under 200
icecream200 <- mean(sales$icecream_sales <200)
icecream200
```

```
[1] 0.3495146
```

```
################ PART 2A
#calculating the expected days of fewer than 200 ice cream sales with a 95% confidence int
binom_test_result <- binom.test(sum(sales$icecream_sales < 200), n = nrow(sales), conf.lev
prop_icecream_less_200 <- binom_test_result$estimate
ci_icecream_less_200 <- binom_test_result$conf.int

#printing the results
binom_test_result
```

```

	Exact binomial test

data:  sum(sales$icecream_sales < 200) and nrow(sales)
number of successes = 36, number of trials = 103, p-value = 0.002929
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.2581794 0.4497513
sample estimates:
probability of success
             0.3495146
```

```
prop_icecream_less_200
```

```
probability of success
             0.3495146
```

```
ci_icecream_less_200
```

```
[1] 0.2581794 0.4497513
attr(,"conf.level")
[1] 0.95
```

```
##########PART 2B
# calculate the total sales
sales$total_sales <- sales$icecream_sales + sales$hotdrink_sales

#calculating the expected days of fewer than 200 TOTAL sales with a 95% confidence interva
binom_test_result_total <- binom.test(sum(sales$total_sales < 200), n = nrow(sales), conf.
prop_total_less_200 <- binom_test_result_total$estimate
ci_total_less_200 <- binom_test_result_total$conf.int

#printing the results
binom_test_result_total
```

```

	Exact binomial test

data:  sum(sales$total_sales < 200) and nrow(sales)
number of successes = 19, number of trials = 103, p-value = 6.208e-11
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.1148697 0.2729811
sample estimates:
probability of success
              0.184466
```

```
prop_total_less_200
```

```
probability of success
              0.184466
```

```
ci_total_less_200
```

```
[1] 0.1148697 0.2729811
attr(,"conf.level")
[1] 0.95
```

```
###############PART 2C
# the odds ratio for a purchase being an ice cream rather than a hot drink in January and
jan <- sales %>% filter(month_name == "Jan")
aug <- sales %>% filter(month_name == "Aug")

# total sales for each category for each month
jan_sales_ice <- sum(jan$icecream_sales)
jan_sales_hot <- sum(jan$hotdrink_sales)
aug_sales_ice <- sum(aug$icecream_sales)
```

```
aug_sales_hot <- sum(aug$hotdrink_sales)

# calculating odds for jan and aug
janodds <- jan_sales_ice / jan_sales_hot
augodds <- aug_sales_ice / aug_sales_hot

# calculating odds ratios
janratio <- janodds
augratio <- augodds

# standard error
se_log_jan <- sqrt(1/jan_sales_ice + 1/jan_sales_hot)
se_log_aug <- sqrt(1/aug_sales_ice + 1/aug_sales_hot)

#95% confidence intervals for log odds ratios
ci_lower_log_jan <- log(janratio) - 1.96 * se_log_jan
ci_upper_log_jan <- log(janratio) + 1.96 * se_log_jan
ci_lower_log_aug <- log(augratio) - 1.96 * se_log_aug
ci_upper_log_aug <- log(augratio) + 1.96 * se_log_aug

# converting back to scale
ci_lower_exp_jan <- exp(ci_lower_log_jan)
ci_upper_exp_jan <- exp(ci_upper_log_jan)
ci_lower_exp_aug <- exp(ci_lower_log_aug)
ci_upper_exp_aug <- exp(ci_upper_log_aug)

#results
janratio
```

```
[1] 0.8676471
```

```
augratio
```

```
[1] 4.117928
```

```
ci_lower_exp_jan
```

```
[1] 0.8080329
```

```
ci_upper_exp_jan
```

```
[1] 0.9316594
```

```
ci_lower_exp_aug
```

```
[1] 3.889579
```

```
ci_upper_exp_aug
```

```
[1] 4.359682
```

```
############# PART 3A
weekdays_sales <- sales[sales$weekend == 0, "total_sales"]
weekends_sales <- sales[sales$weekend == 1, "total_sales"]

t_test_result <- t.test(weekdays_sales, weekends_sales)
t_test_result
```

```

	Welch Two Sample t-test

data:  weekdays_sales and weekends_sales
t = -6.8102, df = 93.836, p-value = 9.171e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -356.0242 -195.2857
sample estimates:
mean of x mean of y 
 324.4038  600.0588 
```

```
######PART 3B
# calculate standard deviation for weekdays and weekends sales
sd_weekdays <- sd(weekdays_sales)
sd_weekends <- sd(weekends_sales)
# calculate the sample size for weekdays and weekends
n_weekdays <- sum(sales$weekend == 0)
```

```
n_weekends <- sum(sales$weekend == 1)

# calculate the pooled standard deviation
sp <- sqrt(((n_weekdays - 1) * sd_weekdays^2 + (n_weekends - 1) * sd_weekends^2) / (n_week
sp
```

```
[1] 204.8695
```

```
# Calculate the effect size
d <- (324 - 600)
d
```

```
[1] -276
```

```
#power level of our data
power.t.test(n = 51.5,
             delta = d, # difference in means
             sd = sp, # pooled standard deviation
             sig.level = 0.05, # significance level
             power = NULL,
             type = "two.sample", # type of t-test
             alternative = "two.sided") # type of alternative hypothesis
```

```

     Two-sample t test power calculation 

              n = 51.5
          delta = 276
             sd = 204.8695
      sig.level = 0.05
          power = 0.9999992
    alternative = two.sided

NOTE: n is number in *each* group
```

```
#########PART 3C
#difference in expected values for 90% power level
power.t.test(n = 51.5,
              delta = -134, # GUESS AND CHECK
              sd = sp, # pooled standard deviation
              sig.level = 0.05, # significance level
              power = NULL,
              type = "two.sample", # type of t-test
              alternative = "two.sided") # type of alternative hypothesis
```

```
     Two-sample t test power calculation

              n = 51.5
          delta = 134
             sd = 204.8695
      sig.level = 0.05
          power = 0.9078088
    alternative = two.sided

NOTE: n is number in *each* group
```

```
############PART 3D
#sample size for a 90% power value
power.t.test(n = 13, #GUESS AND CHECK
              delta = d, # difference in means/sp
              sd = sp, # pooled standard deviation
              sig.level = 0.05, # significance level
              power = NULL,
              type = "two.sample", # type of t-test
              alternative = "two.sided") # type of alternative hypothesis
```

```
     Two-sample t test power calculation

              n = 13
          delta = 276
             sd = 204.8695
      sig.level = 0.05
          power = 0.908959
    alternative = two.sided

NOTE: n is number in *each* group
```

```
##################PART 4A
lm_model <- lm(icecream_sales ~ temperature + humidity + windspeed + weekend + bank_holida
summary(lm_model)
```

```
Call:
lm(formula = icecream_sales ~ temperature + humidity + windspeed + 
    weekend + bank_holiday + school_holidays + month_name, data = sales)

Residuals:
    Min      1Q  Median      3Q     Max 
-121.95  -49.49  -11.14   34.98  184.59 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      -69.4670    50.3776  -1.379   0.1714    
temperature       12.1742     4.5075   2.701   0.0083 ** 
humidity           0.1500     0.4682   0.320   0.7494    
windspeed         -3.1513     1.5560  -2.025   0.0459 *  
weekend1         216.6545    15.4302  14.041  < 2e-16 ***
bank_holiday1    213.3962    36.0012   5.927 5.89e-08 ***
school_holidays1 224.7377    22.7131   9.895 5.92e-16 ***
month_nameAug     68.2961    43.1577   1.582   0.1171    
month_nameFeb    -41.0311    32.6172  -1.258   0.2117    
month_nameJan     17.8839    39.4721   0.453   0.6516    
month_nameJul     55.2285    41.5669   1.329   0.1874    
month_nameJun     57.8862    47.9837   1.206   0.2309    
month_nameMar     42.3650    37.7621   1.122   0.2650    
month_nameMay     26.3115    38.0042   0.692   0.4906    
month_nameSep     29.8036    42.4422   0.702   0.4844    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 71.89 on 88 degrees of freedom
Multiple R-squared:  0.8877,    Adjusted R-squared:  0.8699 
F-statistic: 49.71 on 14 and 88 DF,  p-value: < 2.2e-16
```

```
# creating data frame for prediction a
prediction_data_a <- data.frame(
  temperature = 18,
  humidity = 6,
  windspeed = 10,
  weekend = as.factor("0"),
  bank_holiday = as.factor("0"),
  school_holidays = as.factor("0"),
  month_name = factor("May", levels = levels(sales$month_name))
)

# make predictions
predictions_a <- predict(lm_model, newdata = prediction_data_a, interval = "predict")

# show predictions
head(predictions_a,5)
```

```
       fit       lwr      upr
1 145.3663 -25.36211 316.0948
```

```
meanA <- mean(predictions_a)
meanA
```

```
[1] 145.3663
```

```
# creating data frame for prediction b
prediction_data_b <- data.frame(
  temperature = 28,
  humidity = 35,
  windspeed = 5,
  weekend = as.factor("1"),
  bank_holiday = as.factor("0"),
  school_holidays = as.factor("1"),
  month_name = factor("Apr", levels = levels(sales$month_name))
)

# make predictions
predictions_b <- predict(lm_model, newdata = prediction_data_b, interval = "predict")

# show predictions
head(predictions_b,5)
```

```
       fit      lwr      upr
1 702.2958 470.6987 933.8929
```

```
meanB <- mean(predictions_b)
meanB
```

```
[1] 702.2958
```

```
# creating data frame for prediction c
prediction_data_c <- data.frame(
  temperature = 12,
  humidity = 90,
  windspeed = 35,
  weekend = as.factor("0"),
  bank_holiday = as.factor("0"),
  school_holidays = as.factor("0"),
  month_name = factor("Sep", levels = levels(sales$month_name))
)

# make predictions
predictions_c <- predict(lm_model, newdata = prediction_data_c, interval = "predict")

# show predictions
head(predictions_c,5)
```

```
       fit       lwr      upr
1 9.632756 -176.7343 195.9998
```

```
meanC <- mean(predictions_c)
meanC
```

```
[1] 9.632756
```

```
# creating data frame for prediction d
prediction_data_d <- data.frame(
  temperature = -2,
  humidity = 75,
  windspeed = 15,
  weekend = as.factor("1"),
  bank_holiday = as.factor("0"),
  school_holidays = as.factor("0"),
  month_name = factor("Jan", levels = levels(sales$month_name))
)

# make predictions
predictions_d <- predict(lm_model, newdata = prediction_data_d, interval = "predict")

# show predictions
head(predictions_d,5)
```

```
       fit       lwr      upr
1 104.7053 -65.04169 274.4522
```

```
meanD <- mean(predictions_d)
meanD
```

```
[1] 104.7053
```