Contents

*D*ata Analysis

Tichafa Andrew Rinomhota

Undergraduate in Computer Science, MSU

EXECUTIVE SUMMARY

Consider the pizza delivery data described below: The pizza delivery data is a simulated data set. The data refers to an Italian restaurant which offers home delivery of pizza. It contains the orders received during a period of one month: May 2014. There are three branches of the restaurant. The pizza delivery is centrally managed: an operator receives a phone call and forwards the order to the branch which is nearest to the customer's address. One of the five drivers (two of whom only work part time at the weekend) delivers the order. The data set captures the number of pizzas ordered as well as the final bill which may also include drinks, salads, and pasta dishes. The owner of the business observed an increased number of complaints, mostly because pizzas arrive too late and too cold. To improve the service quality of his business, the owner wants to measure

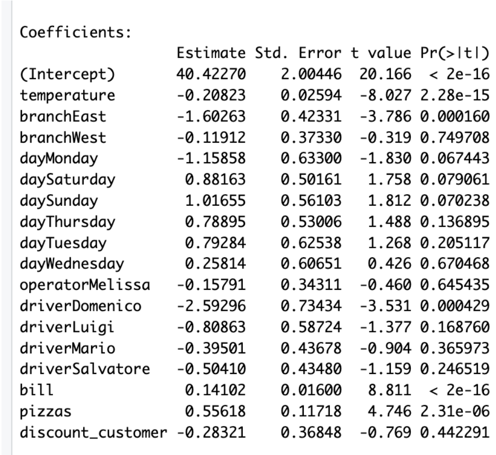
- (i) the time from call to delivery and

- (ii) the pizza temperature at arrival (which can be done with a special device).

  - Ideally, a pizza arrives within 30 min of the call; if it takes longer than 40 min, then the customers are promised a free bottle of water. The temperature of the pizza should be above 65C at the time of delivery. The analysis of the data aims to determine the factors which influence delivery time and temperature of the pizzas.

1 Read the data into R. Fit a multiple linear regression model with delivery time as the outcome and temperature, branch, day, operator, driver, bill, number of ordered pizzas, and discount customer as covariates. Give a summary of the coefficients

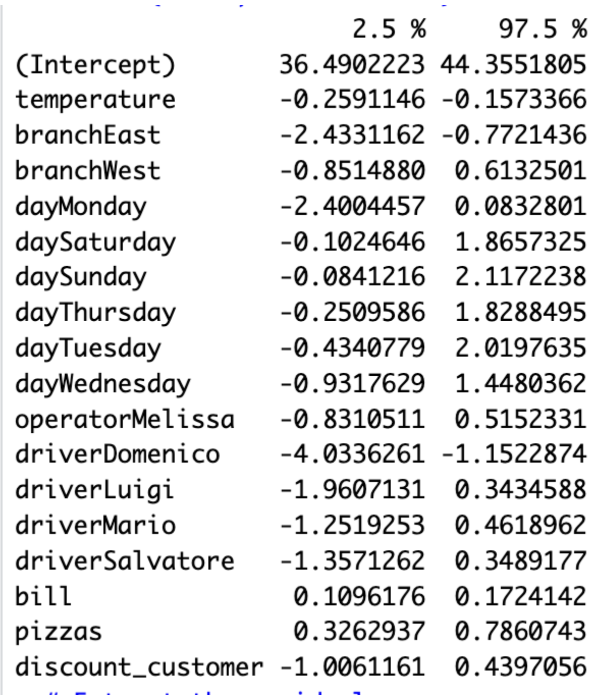
Below are the coefficients and a summary of the multiple linear model. i made sure to fit the multiple linear regression with delivery time as the outcome and temperature, branch, day, operator, driver, bill, number of ordered pizzas, and discount customer as covariates.



```
Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)      40.42270    2.00446  20.166  < 2e-16
temperature      -0.20823    0.02594  -8.027 2.28e-15
branchEast       -1.60263    0.42331  -3.786 0.000160
branchWest       -0.11912    0.37330  -0.319 0.749708
dayMonday        -1.15858    0.63300  -1.830 0.067443
daySaturday       0.88163    0.50161   1.758 0.079061
daySunday         1.01655    0.56103   1.812 0.070238
dayThursday       0.78895    0.53006   1.488 0.136895
dayTuesday        0.79284    0.62538   1.268 0.205117
dayWednesday      0.25814    0.60651   0.426 0.670468
operatorMelissa  -0.15791    0.34311  -0.460 0.645435
driverDomenico   -2.59296    0.73434  -3.531 0.000429
driverLuigi      -0.80863    0.58724  -1.377 0.168760
driverMario      -0.39501    0.43678  -0.904 0.365973
driverSalvatore  -0.50410    0.43480  -1.159 0.246519
bill              0.14102    0.01600   8.811  < 2e-16
pizzas            0.55618    0.11718   4.746 2.31e-06
discount_customer -0.28321    0.36848  -0.769 0.442291
```

2 Use R to calculate the 95 percent confidence intervals of all coefficients

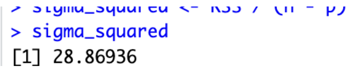
Here is the 95 percent confidence intervals for the coefficents. Returned are the lower and upper bounds of the confidence interval



```
                        2.5 %     97.5 %
(Intercept)       36.4902223 44.3551805
temperature       -0.2591146 -0.1573366
branchEast        -2.4331162 -0.7721436
branchWest        -0.8514880  0.6132501
dayMonday         -2.4004457  0.0832801
daySaturday       -0.1024646  1.8657325
daySunday         -0.0841216  2.1172238
dayThursday       -0.2509586  1.8288495
dayTuesday        -0.4340779  2.0197635
dayWednesday      -0.9317629  1.4480362
operatorMelissa   -0.8310511  0.5152331
driverDomenico    -4.0336261 -1.1522874
driverLuigi       -1.9607131  0.3434588
driverMario       -1.2519253  0.4618962
driverSalvatore   -1.3571262  0.3489177
bill               0.1096176  0.1724142
pizzas             0.3262937  0.7860743
discount_customer -1.0061161  0.4397056
```

3 Reproduce the least squares estimate of sigma squared

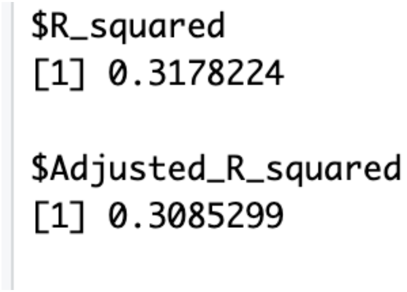
I calculateD the residual variance by dividing the sum of squared residuals (RSS) by (n - p) then this gives me sigma squared.



```
> sigma_squared <- RSS / (n - p)
> sigma_squared
[1] 28.86936
```

4 Now use R to estimate both R2 and R2 adj with the results of model output from part 1. Inter- perate the results

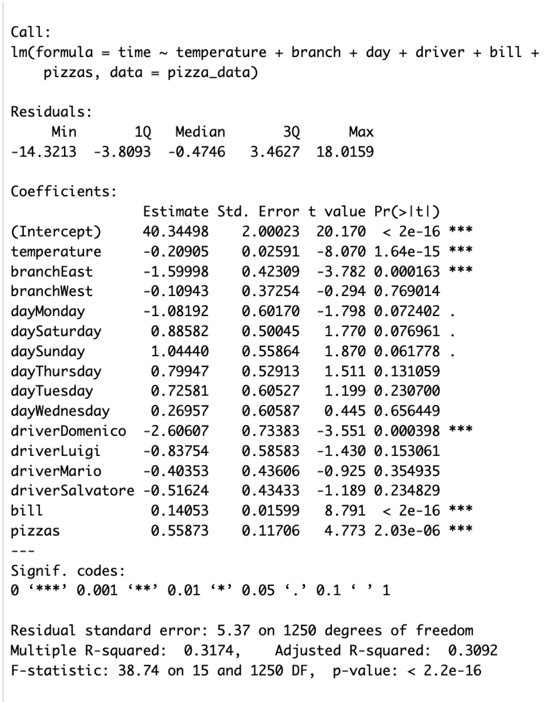
I used R to estimate both R squared and R squared adj. My interpretation* from these results are these results explain how well my model fits. The fact that Adjusted R Squared is slightly lower than R Squared indicates there's most likely no overfitting and that there are predictors.



```
$R_squared
[1] 0.3178224

$Adjusted_R_squared
[1] 0.3085299
```

5 Use backward selection by means of the stepAIC function from the library MASS to find the best model according to AIC

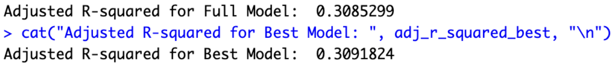
Here I used backward selection by means of the stepAIC from library MASS to find the best model according to AIC and below is the summary of the model.



```
Call:
lm(formula = time ~ temperature + branch + day + driver + bill +
    pizzas, data = pizza_data)

Residuals:
     Min       1Q   Median       3Q      Max
-14.3213  -3.8093  -0.4746   3.4627  18.0159

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     40.34498    2.00023  20.170  < 2e-16 ***
temperature     -0.20905    0.02591  -8.070 1.64e-15 ***
branchEast      -1.59998    0.42309  -3.782 0.000163 ***
branchWest      -0.10943    0.37254  -0.294 0.769014
dayMonday       -1.08192    0.60170  -1.798 0.072402 .
daySaturday      0.88582    0.50045   1.770 0.076961 .
daySunday        1.04440    0.55864   1.870 0.061778 .
dayThursday      0.79947    0.52913   1.511 0.131059
dayTuesday       0.72581    0.60527   1.199 0.230700
dayWednesday     0.26957    0.60587   0.445 0.656449
driverDomenico  -2.60607    0.73383  -3.551 0.000398 ***
driverLuigi     -0.83754    0.58583  -1.430 0.153061
driverMario     -0.40353    0.43606  -0.925 0.354935
driverSalvatore -0.51624    0.43433  -1.189 0.234829
bill             0.14053    0.01599   8.791  < 2e-16 ***
pizzas           0.55873    0.11706   4.773 2.03e-06 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.37 on 1250 degrees of freedom
Multiple R-squared:  0.3174,    Adjusted R-squared:  0.3092
F-statistic: 38.74 on 15 and 1250 DF,  p-value: < 2.2e-16
```

6 Obtain R2 adj from the model identified in 5. and compare it to the full model from 1

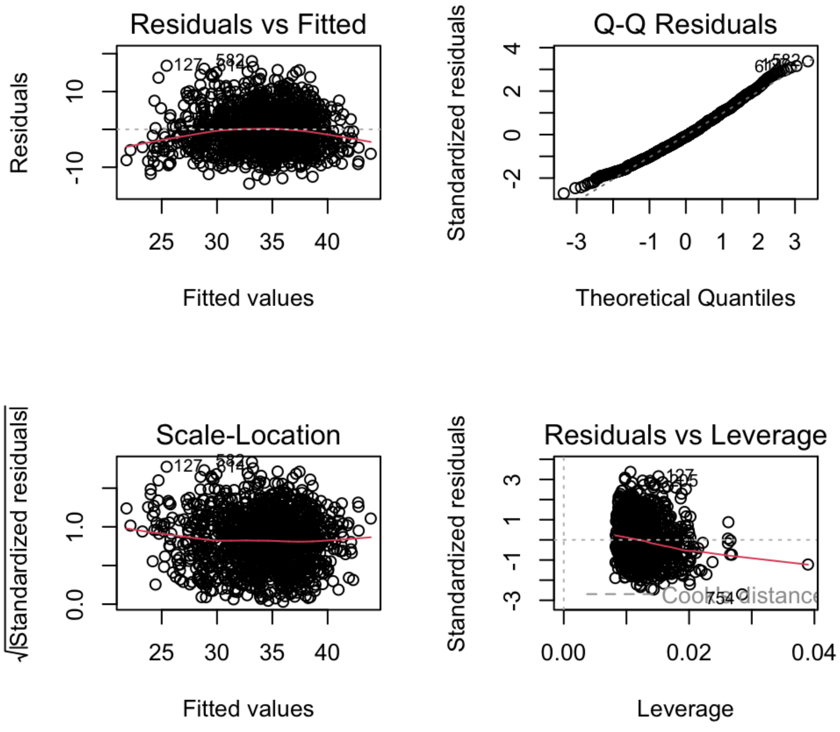
To get the R2 adj from the model in 5 and compare it to the full model from 1, I got Adj R2 from the full model and then Adj R2 from the best model and printed them out.



```
Adjusted R-squared for Full Model:  0.3085299
> cat("Adjusted R-squared for Best Model: ", adj_r_squared_best, "\n")
Adjusted R-squared for Best Model:  0.3091824
```

7 Identify whether the model assumptions are satisfied or not.

To identify if the model assumptions are satisfied or not I plotted the graphs for best model and full model. in residuals vs fittedm, linearity is not really there, but there is normality. It also seems the model is not affected majorly by outliers so I will say the models assumptions are satisfied.

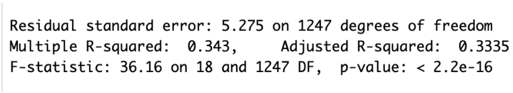


8 Are all variables from the model in 5. causing the delivery time to be either delayed or improved?

I will say based on the summary results from number 5 the variables most of them are negative meaning that delivery time will be improved from my understanding.

9 Test whether it is useful to add a quadratic polynomial of temperature to the model.

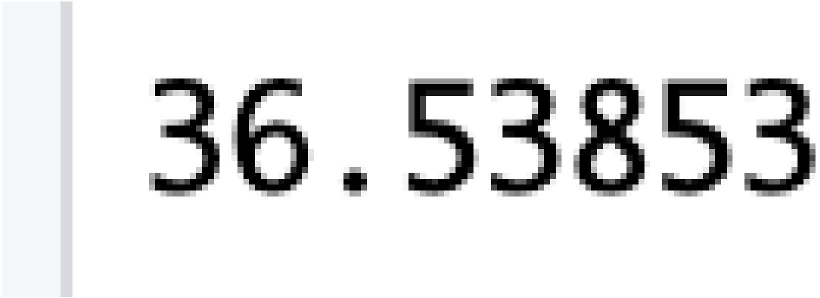
The quadratic polynomial for temperature is useful because it improves the adjusted R-squared of the model. This states that the relationship between temperature and delivery time is not linear.



```
Residual standard error: 5.275 on 1247 degrees of freedom
Multiple R-squared:  0.343,	Adjusted R-squared:  0.3335
F-statistic: 36.16 on 18 and 1247 DF,  p-value: < 2.2e-16
```

10 Use the model identified in 5. to predict the delivery time of the last captured delivery (i.e. number 1266). Use the predict() command to ease the calculation of the prediction.

To come up with predicted time I, took the delivery from the dataset (row 1266) then used the predict() function to calculate the predicted delivery time.



```
36.53853
```

11 Appendix

- R code

```
library(readr)
library(dplyr)


pizza_data <- read_csv('/Users/tichafaandrew/Downloads/pizza_delivery.csv')

model <- lm(DeliveryTime ~ Temperature + Branch + Day + Operator + Driver + Bill + PizzasOrdered + Disc

summary(model)

confint(model, level = 0.95)

residuals <- residuals(model)

RSS <- sum(residuals^2)

 # (n - p)
 n <- nrow(pizza_data)
 p <- length(coefficients(model))


sigma_squared <- RSS / (n - p)
sigma_squared


r_squared <- summary(model)$r.squared
adj_r_squared <- summary(model)$adj.r.squared

list(R_squared = r_squared, Adjusted_R_squared = adj_r_squared)

library(MASS)

#backward selection using stepAIC
best_model <- stepAIC(full_model, direction = "backward")

summary(best_model)

adj_r_squared_full <- summary(full_model)$adj.r.squared

adj_r_squared_best <- summary(best_model)$adj.r.squared

cat("Adjusted R-squared for Full Model: ", adj_r_squared_full, "\n")
cat("Adjusted R-squared for Best Model: ", adj_r_squared_best, "\n")


par(mfrow = c(2, 2))  # Plot 4 graphs in 2x2 layout
plot(full_model)


par(mfrow = c(2, 2))  # Plot 4 graphs in 2x2 layout
plot(best_model)
```

```
best_model <- stepAIC(full_model, direction = "backward")

best_model_with_temp2 <- lm(time ~ temperature + I(temperature^2) + branch + day + operator + driver +


anova(best_model, best_model_with_temp2)

summary(best_model_with_temp2)


best_model <- stepAIC(full_model, direction = "backward")

last_delivery <- pizza_data[1266, ]

predicted_time <- predict(best_model, newdata = last_delivery)

predicted_time
```

12 References

https://r4ds.had.co.nz, https://stackoverflow.com/questions/tagged/r,