Northeastern University
College of Professional Studies

ALY6010

CAPSTONE FINAL PROJECT REPORT

Name: Shreyansh Bhalodiya

NUID : 002664707

Instructor name: Mohsen Soltanifar, PhD, AStat Faculty Lecturer

RESTAURANTS INSPECTION SCORE ANCHORAGE ALASKA

Abstract

This dataset includes a logical Exploratory Data Analysis followed by hypothesis testing and two linear regression models and their interpretations with inspection score as the dependent variable. I have focused more on logical ways of answering the EDA questions.

Key values:

Key variables used: business_name, inspection score, year, Number Of Locations, Weekend.
Test and methods used: Exploratory Data Analysis, Hypothesis Testing, Linear regression models.
Plots used: Bar plot, Scatter plot,Histogram.

1. Introduction

Context:

Inspection scores have been used over the years to maintain the standards of restaurant hygiene and food standards. This scores are inspected each year.

Content:

As we have discussed in milestone 1 and milestone 2 we have selected the dataset "restaurant_inspections" score in Anchorage, Alaska. Let's recap again and then start with our final analysis for this project. Dataset has 27178 rows and 6 columns.

Dataset has the following columns and subjects of the column and its datatype.

- X - Index (Integer)
- Business name - Name of restaurant or name of chain (Character)
- inspection_score - Score at the time of inspection (Integer)
- Year - Year of inspections (Integer)
- Number of locations - number of restaurants for each chain. (Integer)
- Weekends - inspected on weekends or weekdays. False - Weekdays

True - Weekend (Logical)

Acknowledgements

Source of dataset.

https://vincentarelbundock.github.io/Rdatasets/articles/data.html
https://www.kaggle.com/datasets/loulouashley/inspection-score-restaurant-inspection?select=restaurant-and-food-inspections-1.csv

Data Dictionary
https://vincentarelbundock.github.io/Rdatasets/doc/causaldata/restaurant_inspections.html

2.Material and Methods

2.1 Data cleanup

Data cleaning and outlier analysis.

2.2 Exploratory Data Analysis

We have a dataset of restaurant inspection scores performed in Anchorage, Alaska. We have different restaurants and their inspection scores over the years we can describe how the score has improved. Over time also is there any difference if inspection is done on weekends or weekdays how does it affect the score as the number of guests at weekends are more as compared to weekdays. Also we can compare different restaurants based on their number of locations and how inspection went for each of the restaurants over the years. In the previous milestones we have cleaned this dataset and also replaced or removed the null values with logical operations. Now let's start to answer the few questions which we asked in milestone 1.

1. **I would like to see the number of locations for each restaurant over 20 years.**
2. **I will find difference between inspection score at the start of each restaurants and recent years how the trend has changed**
3. **I will also find the age of each restaurant in the span of 20 years.**

2.3 Linear Regression Models

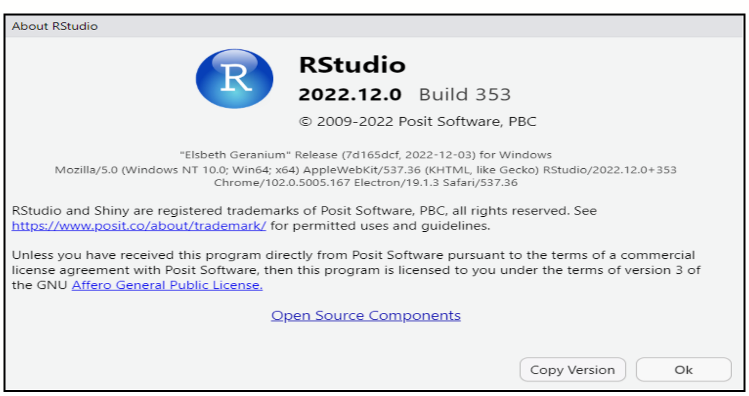
We have used two linear regression models. First is with almost all the variables and secondly with two major variables in the dataset.

Model 1 : Inspection year , inspection score , number of years old , Number of locations.
Model 2: inspection score, number of years old and Number of locations.

2.4 Statistical Software Used

RStudio 2022.12.0

RStudio is an integrated development environment for R, a programming language for statistical computing and graphics. It is available in two formats: RStudio Desktop is a regular desktop application while RStudio Server runs on a remote server and allows accessing RStudio using a web browser.

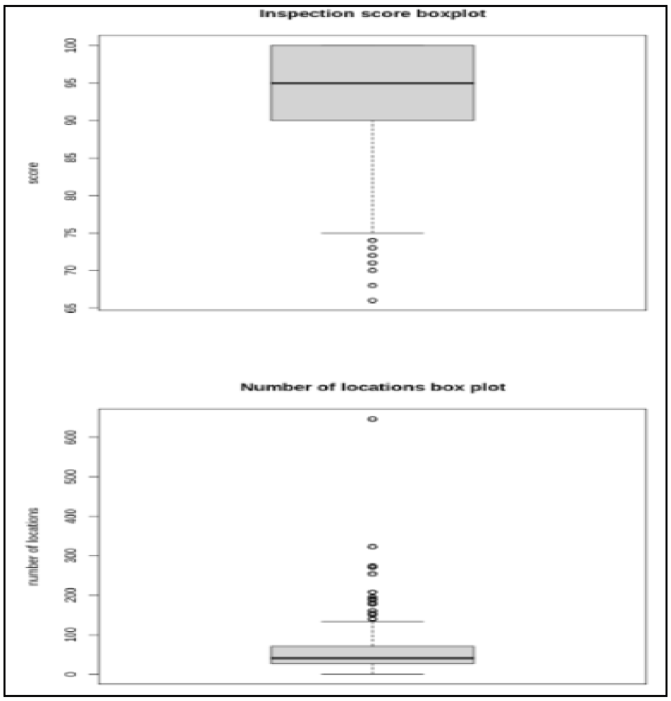


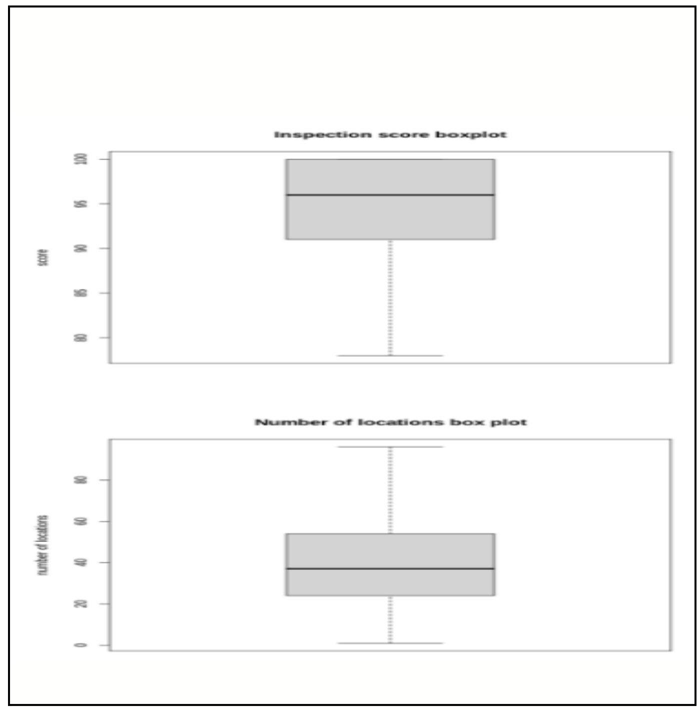
3 Results

3.1 Data Cleaning

● Null value analysis - Data has zero null values hence we can move to the next step for data cleaning

● Outlier Analysis- Here we have two main columns which are integer hence let's check the outlier analysis for those columns using box plot and then using R code let's drop outlier if any and then again plot the box plot. Left Box Plot before dropping outlier and Right after dropping outlier. Hence we have cleaned our data please check R file for coding part of outlier analysis.





3.2 Exploratory Data Analysis

Let's try to find the answers to each of the questions and let's do some deep analysis through step by step coding**.**

Step1: Main Dataframe.

	X	business_name	inspection_score	Year	NumberofLocations	Weekend
	<int>	<chr>	<int>	<int>	<int>	<lgl>
1	1	MCGINLEYS PUB	94	2017	9	FALSE
2	2	VILLAGE INN #1	86	2015	66	FALSE

Step2: Sorting data points for the most recents year for each of the restaurants using library(dplyr)

Results:

A grouped_df: 6 × 4

business_name	Year_recent_yr	inspection_score_recent_yr	NumberofLocations_recent_yr
<chr>	<int>	<int>	<int>
10TH & M SEAFOODS	2018	100	17
12-100 COFFEE & COMMUNITIES	2018	98	11
3 LITTLE PIGS - S	2009	100	19
3M3R LLC DBA YAMA SUSHI	2019	95	13
49TH STATE BREWERY	2017	88	13
49TH STATE BREWERY - BAR	2016	96	3

Step3 :Similarly Sorting data points for the starting year for each of the restaurants

Results:

business_name	Year_start	inspection_score_start	NumberofLocations_start
<chr>	<int>	<int>	<int>
10TH & M SEAFOODS	2009	100	17
12-100 COFFEE & COMMUNITIES	2015	94	11
3 LITTLE PIGS - S	2005	98	19

Step4 : Now let's merge data frames generated in step 2 and step 3 using a left join.

Code:

```
library(tidyverse)

joined_data <- left_join(df_min, df_max, by = "business_name")

joined_data$Location_diff <- joined_data$NumberofLocations_recent_yr-joined_data$NumberofLocations_start

joined_data$num_yrs_old <- joined_data$Year_recent_yr-joined_data$Year_start

joined_data$inspection_score_diff <- joined_data$inspection_score_recent_yr-joined_data$inspection_score_start
```

We are also creating the three new columns here to answer the questions for our analysis. Purpose of creating the new **Location_diff** column will answer the questions related to growth of any restaurants. **Num_yrs_old** will tell the age of each restaurant over the 20 years of span. **Inspection_score_diff** will give an overall idea about the inspection score for each of the restaurants at start and most recent years inspection score difference

Results:

A grouped_df: 6 × 10

business_name	Year_start	inspection_score_start	NumberofLocations_start	Year_recent_yr	inspection_score_recent_yr	NumberofLocations_recent_yr	Location_diff	num_yrs_old	inspection_score_diff
<chr>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
10TH & M SEAFOODS	2009	100	17	2018	100	17	0	9	0
12-100 COFFEE & COMMUNITIES	2015	94	11	2018	98	11	0	3	4
3 LITTLE PIGS - S	2005	98	19	2009	100	19	0	4	2

Analysis:

Looking at the results above and exploring the data further we can see that there are no new locations opened by any store. Possibility of getting such results in data is only the inspection score for each restaurant is updated every year. Other fields in the data are the same as the start of collection of this data. For inspection score difference found that at max the difference is by 23 points also means inspection score difference is less than 0 it means that inspection standards have been maintained by each of the restaurants over the span of 20 years. Furthermore we also found the age of each of the restaurants which can be used as one of the major variables while fitting the regression line.

Code:

```
nf <- data.frame(table(joined_data$num_yrs_old))

barplot(nf$Freq, names.arg=nf$Var1 ,xlab="Year",ylab="Frequency of restaurants number",col="red", main="Frequcy of restaurants being older",border="red",cex.names=0.8)
```

We can see that the majority of frequency is for either older restaurants and new restaurants. Overall there are more new restaurants opened in less than 3 years and more restaurants for more than17 years. We can create three clusters for 0-3 years, 4-16 years and 17-19 years.

3.3 Linear Regression

Creating Dummies for Regression lines.

```
install.packages("fastDummies")
library('fastDummies')
```

https://www.marsja.se/create-dummy-variables-in-r/

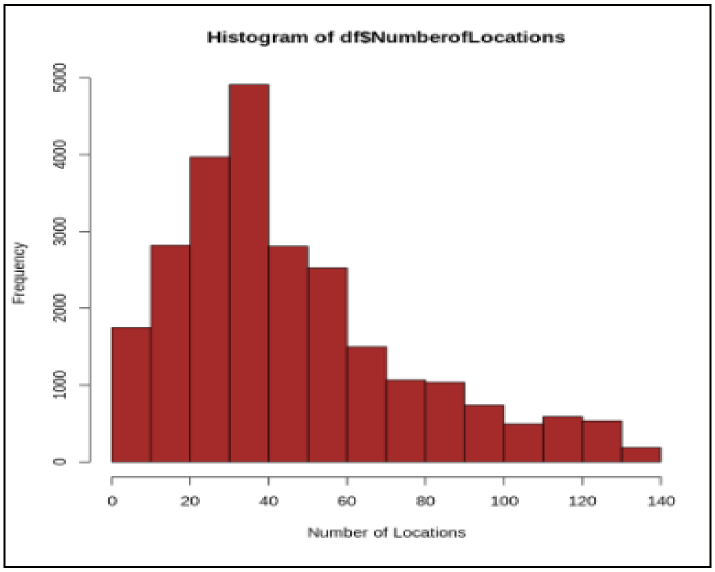
A data.frame: 6 × 8

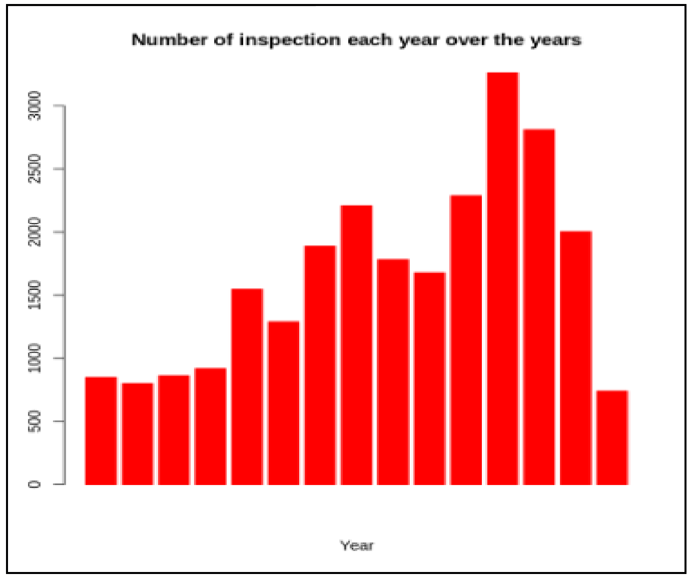
X	business_name	inspection_score	Year	NumberofLocations	Weekend	count	num_yrs_old
<int>	<chr>	<int>	<chr>	<int>	<lgl>	<dbl>	<int>
1	MCGINLEYS PUB	94	2017	9	FALSE	1	10
2	VILLAGE INN #1	86	2015	66	FALSE	1	19
3	RONNIE SUSHI 2	80	2016	79	FALSE	1	1
4	FRED MEYER - RETAIL FISH	96	2003	86	FALSE	1	19
5	PHO GRILL	83	2017	53	FALSE	1	3
6	TACO KING #2	95	2008	89	FALSE	1	16

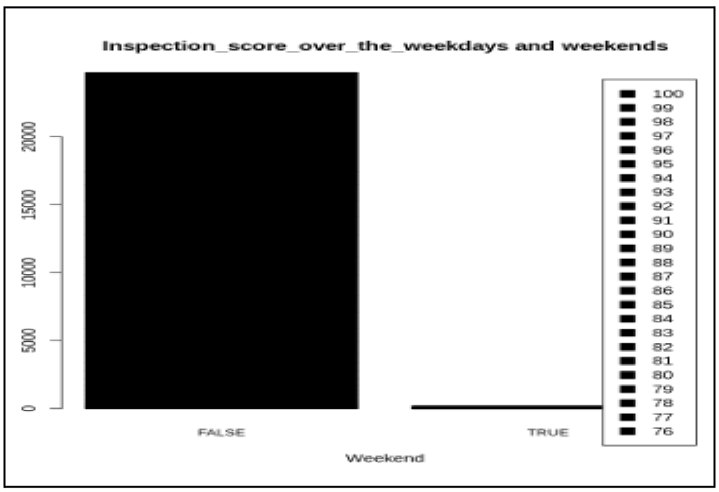
We found a num_yrs_old variable from above questions and I have joined with my main dataframe and will use this for my linear regression model. For creating the dummy year, num_yrs_old and weekend will use Number_of_locations to feed as it is.

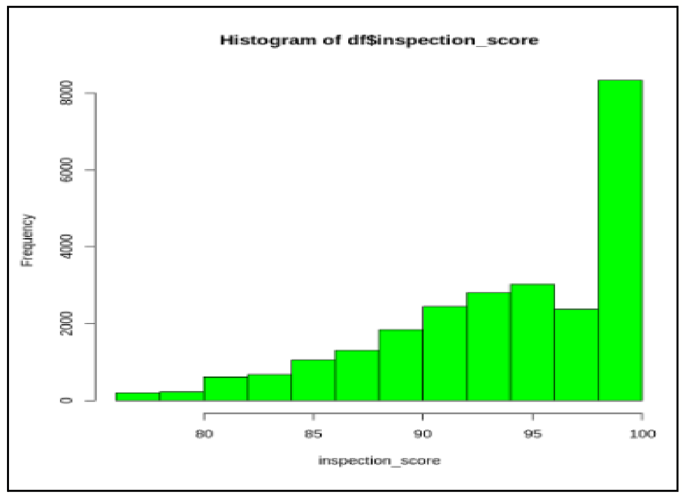
num_yrs_old	Year_2000	Year_2001	⋯	num_yrs_old_12	num_yrs_old_13	num_yrs_old_14	num_yrs_old_15	num_yrs_old_16	num_yrs_old_17	num_yrs_old_18	num_yrs_old_19	Weekend_FALSE	Weekend_TRUE
<chr>	<int>	<int>	⋯	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
10	0	0	⋯	0	0	0	0	0	0	0	0	1	0
19	0	0	⋯	0	0	0	0	0	0	0	1	1	0
1	0	0	⋯	0	0	0	0	0	0	0	0	1	0
19	0	0	⋯	0	0	0	0	0	0	0	1	1	0
3	0	0	⋯	0	0	0	0	0	0	0	0	1	0
16	0	0	⋯	0	0	0	0	1	0	0	0	1	0

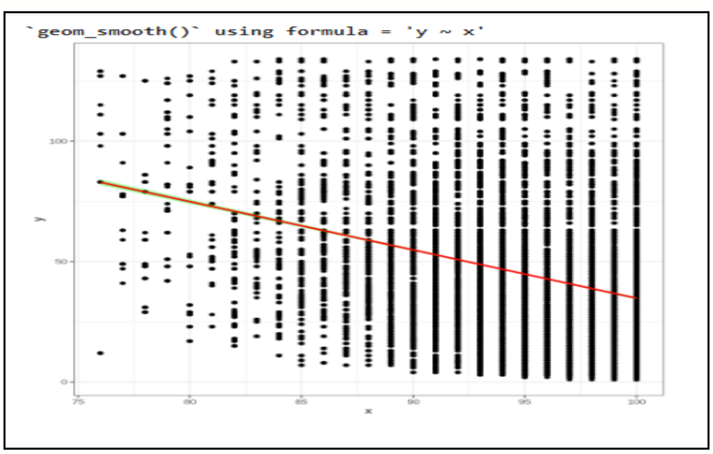
Before fitting the regression line, let's first try to fit the regression line with major variables and try to visualize our results.

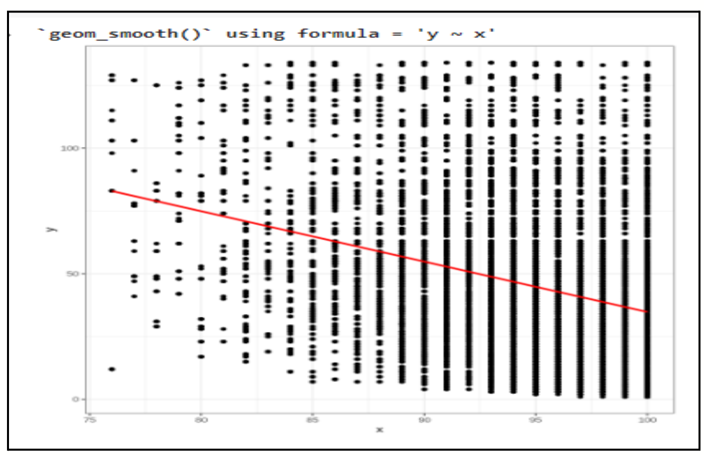












- On the left we have variable weekends. Looking at the graph it is clear that weekends have very low inspection days and major inspection days are on weekdays hence learning from this variable is very less.

- On the right we have a regression line with a scatter plot between inspection score and Number of locations. But looking at the plot we can see that there is negative correlation between two variables but no major learning is there for fitting regression as data is not showing any trends.

MODEL 1

```
model = lm(inspection_score~NumberofLocations +
    Year_2000+Year_2001+Year_2002+Year_2003+Year_2004+Year_2005+Year_2006+Year_2007+Year_2008+Year_2009
    +Year_2015+Year_2016+

    Year_2017+Year_2018+Year_2019+num_yrs_old_0+num_yrs_old_1+num_yrs_old_2+num_yrs_old_3+num_yrs_old_4
    +num_yrs_old_6+num_yrs_old_7+num_yrs_old_8+num_yrs_old_9+num_yrs_old_10+num_yrs_old_11+

    num_yrs_old_12+num_yrs_old_13+num_yrs_old_14+num_yrs_old_15+num_yrs_old_16+num_yrs_old_17+num_y
    rs_old_18+num_yrs_old_19+Weekend_FALSE+Weekend_TRUE
  , data = dataf)
#Create a linear regression with multiple variables.
```

Results

```
Call:
lm(formula = inspection_score ~ NumberofLocations + Year_2000 +
    Year_2001 + Year_2002 + Year_2003 + Year_2004 + Year_2005 +
    Year_2006 + Year_2007 + Year_2008 + Year_2009 + Year_2015 +
    Year_2016 + Year_2017 + Year_2018 + Year_2019 + num_yrs_old_0 +
    num_yrs_old_1 + num_yrs_old_2 + num_yrs_old_3 + num_yrs_old_4 +
    num_yrs_old_6 + num_yrs_old_7 + num_yrs_old_8 + num_yrs_old_9 +
    num_yrs_old_10 + num_yrs_old_11 + num_yrs_old_12 + num_yrs_old_13 +
    num_yrs_old_14 + num_yrs_old_15 + num_yrs_old_16 + num_yrs_old_17 +
    num_yrs_old_18 + num_yrs_old_19 + Weekend_FALSE + Weekend_TRUE,
    data = dataf)

Residuals:
    Min      1Q  Median      3Q     Max
-21.136  -2.910   0.866   3.404  15.054

Coefficients: (3 not defined because of singularities)
                Estimate Std. Error t value Pr(>|t|)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	101.87609	0.390573	260.838	< 2e-16 ***
NumberofLocations	-0.095004	0.001161	-81.801	< 2e-16 ***
Year_2000	0.062381	0.248403	0.251	0.801718
Year_2001	0.492893	0.251082	1.963	0.049648 *
Year_2002	0.605432	0.247284	2.448	0.014359 *
Year_2003	1.201351	0.24391	4.925	8.47e-07 ***
Year_2004	-0.113833	0.222777	-0.511	0.609374
Year_2005	-0.714245	0.228965	-3.119	0.001814 **
Year_2006	-1.942391	0.216183	-8.985	< 2e-16 ***
Year_2007	-1.328040	0.21206	-6.263	3.85e-10 ***
Year_2008	-1.414606	0.217735	-6.497	8.35e-11 ***
Year_2009	-1.211318	0.219986	-5.506	3.70e-08 ***
Year_2015	0.742392	0.206558	3.594	0.000326 ***
Year_2016	-0.509305	0.199159	-2.557	0.010555 *
Year_2017	-0.013535	0.201881	-0.067	0.946545
Year_2018	-0.175772	0.20875	-0.842	0.399786
Year_2019	NA	NA	NA	NA

	Estimate	Std. Error	t value	Pr(>|t|)
num_yrs_old_0	-6.817285	0.231838	-29.405	< 2e-16 ***
num_yrs_old_1	-6.922776	0.186875	-37.045	< 2e-16 ***
num_yrs_old_2	-6.582100	0.153359	-42.92	< 2e-16 ***
num_yrs_old_3	-5.459087	0.1562	-34.949	< 2e-16 ***
num_yrs_old_4	-4.752510	0.218677	-21.733	< 2e-16 ***
num_yrs_old_6	-4.045972	2.761971	-1.465	0.142965
num_yrs_old_7	-2.433157	0.688735	-3.533	0.000412 ***
num_yrs_old_8	-5.031154	0.323702	-15.543	< 2e-16 ***
num_yrs_old_9	-5.317053	0.29008	-18.33	< 2e-16 ***
num_yrs_old_10	-4.836817	0.296437	-16.317	< 2e-16 ***
num_yrs_old_11	-2.066021	0.262726	-7.864	3.88e-15 ***
num_yrs_old_12	-2.361502	0.273015	-8.650	< 2e-16 ***
num_yrs_old_13	-2.974407	0.221708	-13.416	< 2e-16 ***
num_yrs_old_14	-1.528395	0.226766	-6.740	1.62e-11 ***
num_yrs_old_15	-0.727565	0.179027	-4.064	4.84e-05 ***
num_yrs_old_16	-1.701063	0.14013	-12.139	< 2e-16 ***
num_yrs_old_17	-2.014779	0.099149	-20.321	< 2e-16 ***
num_yrs_old_18	-1.591670	0.097014	-16.407	< 2e-16 ***
num_yrs_old_19	NA	NA	NA	NA
Weekend_FALSE	-0.337494	0.338557	-0.997	0.318842
Weekend_TRUE	NA	NA	NA	NA

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.776 on 24888 degrees of freedom
Multiple R-squared:  0.2768,	Adjusted R-squared:  0.2758
F-statistic: 280.2 on 34 and 24888 DF,  p-value: < 2.2e-16
```

MODEL 2

model_2 = lm(inspection_score~NumberofLocations+num_yrs_old, data = dataf) #Create a ## linear regression with two variables
summary(model_2)

Call:
lm(formula = inspection_score ~ NumberofLocations + num_yrs_old,
data = dataf)
Residuals:

Min	1Q	Median	3Q	Max
-20.2645	-3.0115	0.9998	3.3171	13.8333

(Intercept)	94.59489	0.208519	453.65	< 2.00E-16**
NumberofLocatio	-0.097599	0.001165	-83.797	< 2e-16 ***
num_yrs_old1	-0.111193	0.263484	-0.422	0.6730
num_yrs_old10	1.612107	0.357319	4.512	6.46e-06 ***
num_yrs_old11	4.326034	0.332830	12.998	< 2e-16 ***
num_yrs_old12	3.926977	0.341058	11.514	< 2e-16 ***
num_yrs_old13	3.267995	0.301867	10.826	< 2e-16 ***
num_yrs_old14	4.825028	0.303689	15.888	< 2e-16 ***
num_yrs_old15	5.825856	0.270606	21.529	< 2e-16 ***
num_yrs_old16	4.804486	0.247968	19.375	< 2e-16 ***
num_yrs_old17	4.552454	0.226891	20.064	< 2e-16 ***
num_yrs_old18	5.015928	0.225948	22.199	< 2e-16 ***
num_yrs_old19	6.671183	0.225656	29.563	< 2e-16 ***
num_yrs_old2	0.333166	0.242035	1.377	0.1687
num_yrs_old3	1.552072	0.244501	6.348	2.22e-10 ***
num_yrs_old4	2.285911	0.291421	7.844	4.54e-15 ***
num_yrs_old6	2.478694	2.803162	0.884	0.3766
num_yrs_old7	4.152980	0.723003	5.744	9.35e-09 ***
num_yrs_old8	1.557108	0.381855	4.078	4.56e-05 ***
num_yrs_old9	1.130680	0.353869	3.195	0.0014 **

Signif. codes: 0 '*' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1**

Residual standard error: 4.842 on 24903 degrees of freedom
Multiple R-squared: 0.2562, Adjusted R-squared: 0.2556
F-statistic: 451.4 on 19 and 24903 DF, p-value: < 2.2e-16

4.Discussion

4.1 Modeling:

Output represents the results of a linear regression model fitted to the data in dataf. Model is used to predict the inspection_score based on a number of predictor variables, including NumberofLocations, Year_2000, Year_2001, Year_2002, Year_2003, Year_2004, Year_2005, Year_2006, Year_2007, Year_2008, Year_2009, Year_2015, Year_2016, Year_2017, Year_2018, Year_2019, num_yrs_old_0, num_yrs_old_1, num_yrs_old_2, num_yrs_old_3, num_yrs_old_4, num_yrs_old_6, num_yrs_old_7, num_yrs_old_8, num_yrs_old_9, num_yrs_old_10, num_yrs_old_11, num_yrs_old_12, num_yrs_old_13, num_yrs_old_14, num_yrs_old_15, num_yrs_old_16, num_yrs_old_17, num_yrs_old_18, num_yrs_old_19, Weekend_FALSE, and Weekend_TRUE.

The Coefficient table shows the estimated effect of each predictor on the inspection_score outcome variable. The Estimate column gives the estimated effect of each predictor variable on the outcome variable. The Std. Error column gives the standard error of the estimate, which indicates how much the estimate is likely to vary from the true value. The t value column gives the value of the t-statistic for testing the null hypothesis that the true effect of the predictor variable on the outcome variable is zero. Finally, the Pr(>|t|) column gives the p-value for this test, which represents the probability of observing a t-statistic at least as extreme as the observed value if the null hypothesis is true. If the p-value is less than a specified level (usually 0.05), then we reject the null hypothesis and conclude that the predictor variable has a significant effect on the outcome variable. The intercept term (which has no predictor variable associated with it) represents the estimated mean value of the outcome variable when all the predictor variables are zero.

Overall we tried two linear regression models with two different sets of variables. In model 1 it is clear that even after using all the major variables r value is just 0.27 which is a low score. Similarly for model 2 we tried to predict inspection score using two variables. We don't find any major trends in data as major variables which are used in calculating the inspection score are missing. Also data is narrow for fitting any model as it is limited to few variables.

4.2 EDA

In Exploratory data analysis we majorly address three questions first is the number of locations when restaurants started and recently. We found that the difference between those locations is zero. Hence either data was not updated or inspection was done in the same location for over the 20 years. Furthermore in the similar lines if we analyze the inspection score trends it has been at max difference of 23 points and on average less than 1 points hence overall restaurants were successful in maintaining their inspection score over the 20 years of span. Another possibility is that inspection is conducted over the same data points over the 20 years and restaurants were smart enough to keep good scores particularly for those data points. Frequency of restaurants being older than 3 years is highest followed by 17-19 years and finally the rest of the years between 4-16. Overall dataset we selected was impactful for descriptive analysis. For inferential analysis we needed more variables to find major trends and predict the inspection score.

4.3 Limitations

Dataset selected was limited to descriptive analysis as a limited number of columns were available for analysis. Main subject column in data was inspection score and no major major variables were available which had very high correlation to variable inspection score. Data was more general and informative for restaurants which is generally used for record keeping.

4.4 Future Works

Data was excellent for descriptive analysis but was more of record keeping and not fit for analysis. But this type of data if maintained properly can be used to automate things like predicting the inspection score before inspection. This can help the restaurants to take the required steps before inspection. In future if we get more variables we can build an accurate algorithm with good R-value and can deploy it and can help restaurants to predict their inspection scores.

References

- *Available datasets*. (n.d.). https://vincentarelbundock.github.io/Rdatasets/articles/data.html
- *R: Data on Restaurant Inspections*. (n.d.). https://vincentarelbundock.github.io/Rdatasets/doc/causaldata/restaurant_inspections.html
- Johnson, D. (2023, March 25). *T-Test in R Programming: One Sample & Paired T-Test [Example]*. Guru99. https://www.guru99.com/r-t-test-one-sample.html
- *Unpaired Two-Samples T-test in R - Easy Guides - Wiki - STHDA*. (n.d.). http://www.sthda.com/english/wiki/unpaired-two-samples-t-test-in-r
- Marsja, E. (2021, April 15). *How to Create Dummy Variables in R (with Examples)*. Erik Marsja. https://www.marsja.se/create-dummy-variables-in-r/

Appendix

```
url_next <- "https://vincentarelbundock.github.io/Rdatasets/csv/causaldata/restaurant_inspections.csv"
df <- read.csv(url_next)

## Let's first understand the dataframe columns and its subject and get basic idea of what data is regarding.

## Business name - Name of restaurent or name of chain
## inspection_score - Score at the time of inspection
## Year - Year of inspections
## Number of locations of restaurent of particular location.
## was inspection done on week end or weekdays. False - Weekdays True - Weekend.

## data source - https://vincentarelbundock.github.io/Rdatasets/articles/data.html  &
https://www.kaggle.com/datasets/loulouashley/inspection-score-restaurant-inspection?select=restaurant-and-food-inspections-1.csv
## data dictionary - https://vincentarelbundock.github.io/Rdatasets/doc/causaldata/restaurant_inspections.html

head(df)

## purpose of dataset - We have dataset of restaurent inspection score performed in Anchorage, Alaska. We have different restaurents and
their inspection scores over the years we can describle that how the score has imporoved
## over the time also is there any difference if inspection done in weekends or weekdays how does it affect the score as number of guest
at weekends are more as compare to weekdays.

dim(df)
str(df)

## Let's check do we need to drop any columns or replace null values with any desired number ?

sum(is.na(df))

## we can clearly see that data has zero null values.

##  let's print box plot for outlier analysis.

bx_1 <- boxplot(df$inspection_score,
             main = "Inspection score boxplot",
             ylab = "score")

bx_2 <- boxplot(df$NumberofLocations,
             main = "Number of locations box plot",
```

```
            ylab = "number of locations")


## we can see that we have few outliers for inspection score column in lower quartiles and few outliers in upper quartiles for number of locations..


q1 <- quantile(df$inspection_score, 0.25)
q3 <- quantile(df$inspection_score, 0.75)
IQR <- q3-q1
lower <- q1 - 1.5*IQR
upper <- q3 + 1.5*IQR
df <- df[ which(df$inspection_score < upper
            & df$inspection_score > lower), ]
cat(lower,upper)

q1 <- quantile(df$NumberofLocations, 0.25)
q3 <- quantile(df$NumberofLocations, 0.75)
IQR <- q3-q1
lower <- q1 - 1.5*IQR
upper <- q3 + 1.5*IQR
df <- df[ which(df$NumberofLocations < upper
            & df$NumberofLocations > lower), ]
cat(lower,upper)

## lower and upper for inspection column is 77.5 and 113.5 we can clealry see in box plot all data point are in given range hence no outliers.
## lower and upper for numberof location column is -20 and 100 we can clealry see in box plot all data point are in given range hence no outliers.
## we can clearly see that we have removed outliers from both the columns and now our data is cleaned.
bx_1_no_out <- boxplot(df$inspection_score,
                  main = "Inspection score boxplot",
                  ylab = "score" )

bx_2_no_out <- boxplot(df$NumberofLocations,
                  main = "Number of locations box plot",
                  ylab = "number of locations")

### data is free from null value and outliers let's explore further.

names <- data.frame(table(df$business_name))
names <- setNames(names, c("name","count"))
names <-names[order(names$stores,decreasing=TRUE,na.last=FALSE),]
head(names,10)

## we can clearly see that there are few restaurents which are repeating so let's check how many are unique.

dim(names)

## so out of 27178 number of uniques names is 1571 hence we can clearly see that each year inspection is conducted and same is recored in this dataframe.

## let's check summary of cleaned dataframe.
## here we can see that data is for 20 years starting from 2000 to 2019.
## Also mostly inspection is conducted on weekdays.
## max inspection score in cleaned data is 100 i.e ideal number of scoring range and maximum number locations for any store is 96.
summary(df)


hist(df$NumberofLocations,xlab = "Number of Locations",col = "brown",border = "black")
```

```
## we can see that as number of locations increases in x axis frequency of number of stores with more number of locations decreases which is the ideal situation in real time.

hist(df$inspection_score,xlab = "inspection_score",col = "green",border = "black")
### we can clealry see that maximum number of restaurents hold inspection score which is greater then 95 which is actually a positive score and good score for those restaurants.

df["count"] <- 1
head(df)

## If we compare inspection with years we get an idea each year how many inspections were done over these 20 years and also average number of inspection scores each year and how the score is shifting over these 20 years.

## First let's find how many inspection were done each year.
g_yer <- data.frame(aggregate(df$count, list(df$Year), FUN=sum))
g_yer <- setNames(g_yer, c("year","Num_of_inspection"))
g_yer


barplot(g_yer$Num_of_inspection, names.arg=g_yer$Year ,xlab="Year",ylab="Number of inspection",col="red", main="Number of inspection each year over the years",border="red")
mean(g_yer_score$inspection_score_avg)

## Now let's findaverage inspection score each year.
g_yer_score <- data.frame(aggregate(df$inspection_score, list(df$Year), FUN=mean))
g_yer_score <- setNames(g_yer_score, c("year","inspection_score_avg"))
g_yer_score


barplot(g_yer_score$inspection_score_avg, names.arg=g_yer_score$Year ,xlab="Year",ylab="inspection_score_avg",col="blue", main="average score of inspection each year over the years",border="red")


## ONE SAMPLE T TEST
## H0 NULL : Average mean of average of each year is 96
## H1 ALTERNATE : mean is not equal to 96


## https://www.guru99.com/r-t-test-one-sample.html


t.test(g_yer_score$inspection_score_avg, mu = 96)

head(df)
str(df)


### Looking at the above graph we can see that average mean of inspection score in year 2018 and 2017 is same
## let's see that same with two sample t-test.

df_17 = df[df["Year"] == 2017,]
df_18 = df[df["Year"] == 2018, ]

library(dplyr)
#df_17 %>% filter(row("Year") == 2017)

## http://www.sthda.com/english/wiki/unpaired-two-samples-t-test-in-r
two_tailed <- t.test(df_17['inspection_score'], df_18['inspection_score'])
two_tailed
head(df)
```

```
library(dplyr)
df_max <- df %>%
  group_by(business_name) %>%
  slice(which.max(Year))
column <- c("business_name","Year","inspection_score","NumberofLocations")
df_max <- df_max[column]
names(df_max)[names(df_max) == "inspection_score"] <- "inspection_score_recent_yr"
names(df_max)[names(df_max) == "Year"] <- "Year_recent_yr"
names(df_max)[names(df_max) == "NumberofLocations"] <- "NumberofLocations_recent_yr"
head(df_max)


library(dplyr)
df_min <- df %>%
  group_by(business_name) %>%
  slice(which.min(Year))
column <- c("business_name","Year","inspection_score","NumberofLocations")
df_min <- df_min[column]
names(df_min)[names(df_min) == "inspection_score"] <- "inspection_score_start"
names(df_min)[names(df_min) == "Year"] <- "Year_start"
names(df_min)[names(df_min) == "NumberofLocations"] <- "NumberofLocations_start"
head(df_min)


library(tidyverse)
joined_data <- left_join(df_min, df_max, by = "business_name")

dim(joined_data)

joined_data$Location_diff <- joined_data$NumberofLocations_recent_yr-joined_data$NumberofLocations_start
joined_data$num_yrs_old <- joined_data$Year_recent_yr-joined_data$Year_start
joined_data$inspection_score_diff <- joined_data$inspection_score_recent_yr-joined_data$inspection_score_start

head(joined_data)
dim(joined_data)

joined_data <- joined_data[order(-joined_data$inspection_score_diff),]
head(joined_data)
mean(joined_data$inspection_score_diff)

nf <- data.frame(table(joined_data$num_yrs_old))
barplot(nf$Freq, names.arg=nf$Var1 ,xlab="Year",ylab="Frequency of restaurants number",col="red", main="Frequcy of restaurants
being older",border="red",cex.names=0.8)
head(df)

install.packages("fastDummies")
library('fastDummies')

### https://www.marsja.se/create-dummy-variables-in-r/
df$Year <- as.character(df$Year)

library(tidyverse)
column <- c("business_name","num_yrs_old")
df_zz <- joined_data[column]
new_frame <- left_join(df, df_zz, by = "business_name")

head(new_frame)
new_frame$num_yrs_old <- as.character(new_frame$num_yrs_old)

head(new_frame)
```

```
new_frame$num_yrs_old <- as.character(new_frame$num_yrs_old)

dataf <- dummy_cols(new_frame, select_columns = c('Year', 'num_yrs_old','Weekend'))

model = lm(inspection_score~NumberofLocations +
Year_2000+Year_2001+Year_2002+Year_2003+Year_2004+Year_2005+Year_2006+Year_2007+Year_2008+Year_2009+Year_2015+Ye
ar_2016+

Year_2017+Year_2018+Year_2019+num_yrs_old_0+num_yrs_old_1+num_yrs_old_2+num_yrs_old_3+num_yrs_old_4+num_yrs_old_6
+num_yrs_old_7+num_yrs_old_8+num_yrs_old_9+num_yrs_old_10+num_yrs_old_11+

num_yrs_old_12+num_yrs_old_13+num_yrs_old_14+num_yrs_old_15+num_yrs_old_16+num_yrs_old_17+num_yrs_old_18+num_yrs
_old_19+Weekend_FALSE+Weekend_TRUE
          , data = dataf)
#Create a linear regression with multiple variables.
summary(model)

new_frame$num_yrs_old <- as.numeric(new_frame$num_yrs_old)
model_2 = lm(inspection_score~NumberofLocations+num_yrs_old, data = new_frame) #Create a linear regression with two variables
summary(model_2) #Review the result

cor(df$inspection_score,df$NumberofLocations)

install.packages("ggplot2")
library(ggplot2)

counts <- table(df$inspection_score,df$Weekend)
barplot(counts, main="Inspection_score_over_the_weekdays and weekends",
        xlab="Weekend", col="black",
        legend = rownames(counts),cex.names=0.8, beside=FALSE)

x <- dataf$inspection_score
y <- dataf$NumberofLocations
plt <- ggplot(dataf, aes(x=x,y=y)) + geom_point(color="black")+theme_bw()
plt

plt2 <- plt + geom_smooth(method = lm, color="red",se=FALSE)
plt2

plt3 <- plt + geom_smooth(method = lm , color = "red", fill="green",se= TRUE)
plt3
```