MathWorks Math Modeling Challenge 2022

New Century Technology High School

Team #15559, Huntsville, Alabama

Coach: Clifford Pate

Students: Shreyas Puducheri, Donal Higgins, Alexander Ivan, Ella Duus

M3 Challenge Technical Computing Award
THIRD PLACE—$1,000 Team Prize

JUDGE COMMENTS

Specifically for Team # 15559 —Submitted at the Close of Technical Computing Contention Judging:

COMMENT 1: This paper effectively used a notebook coding environment (Python Jupyter Notebooks) to perform almost all of the calculations and modeling included in the paper. Code was well commented, and the notebook allowed the team to easily sanity check and display intermediate results. Like other top TC teams, this paper employed a machine learning model (a random forest classifier) for Q2, which was trained on real survey data. The judges were impressed with how that data was processed and cleaned automatically using code. We were also impressed with the team's thoughtful approach to understanding the trained machine learning model. They leveraged the "feature importance" function from the Scikit-learn library used to train the model, which helped them understand what demographic data is most important in driving the decision to work from home. One weakness of the paper was data visualizations – for example, we would have liked to see a more concise representation of the regression lines constructed for Q1. We would also have liked to see more effective code-reuse and less hard-coding of parameters. Overall though, this was a strong paper, which serves as a good example for how technical computing can be used to structure and organize the entire modeling process from beginning to end.

Specifically for Team #15559 —Submitted at the Close of Triage Judging:

COMMENT 1: The team has presented a simple and clean model for Q1 that is quite easily reproduceable. The validity checks of the results are also decent. For Q2 (and consequently Q3), it would have been good, if the team had had more time, to document more the random forest algorithm and how they used it, to allow both reproduceability and model checking.

COMMENT 2: Well done. Good explanations of your methods and results.

COMMENT 3: Model for Q1 explained very clearly

COMMENT 4: It would be appropriate to demonstrate that a linear relationship would be most appropriate to model the data. Perhaps displaying a scatterplot to justify a linear model. There were some statements as to the coefficient of determination and the correlation coefficient and their interpretation in the context of a linear relationship that were a little concerning. I liked the Data Science approach in question 2. Be careful, the question 2 and question 3 stated were identical in you report.

Remote Work: *Fad or Future*

0 Executive Summary

When the COVID-19 pandemic disrupted every avenue of life, millions of working individuals adjusted through telecommuting. Now, the question is whether remote work will remain a part of the new normal. Our team's goal is to model the trajectory of remote work in select cities for 2024 and 2027, considering pertinent factors such as the availability of remote-ready jobs, employer decisions, and employee choices.

To begin, we calculated the maximum number of individuals in a city who could feasibly work remotely in their current job position. In other words, we wanted to find the "remote-ready" work population for each city. To do so, we multiplied the percentage of individuals in each industry who could work from home by the predicted number of individuals in a given industry over the period 2022-2027. To predict the latter value, we performed linear regression on the provided M3 Mathworks data for the number of individuals in each industry in each city over time. We thus predicted the remote-ready population for each city for the years 2024 and 2027. Despite having some low R2 values—the year was sometimes a poor predictor of the number of individuals in an industry—the model that combined numerous linear regressions yielded reasonable results for the remote-ready populations for the years 2024 and 2027.

Furthermore, to predict whether an individual worker will choose to work remotely full-time and gain employer approval, we created a random forest classification based on relevant factors to the employee choice such as age, gender, and parental status. This yielded a relatively good accuracy of 0.74, and we identified age as the most important factor in the worker's decision to work remotely or in person. We accounted for the employer's likelihood to allow the employee to work remotely with a random probability generation.

Finally, to use our model that predicted whether both the employer and a given individual would agree to work remotely for the five cities, we simulated 5000 citizens of each city. The simulation accounted for the unique distribution of ages, genders, and children for each city and provided us with the percent of employees and employers who would agree on fully remote working plans, given that the individual's job could be completed remotely. Combining these 5 percentages for employee and employer cooperation, we have found predictions for all 3 years for all 5 cities for the number of individuals who will work remotely. We have found that Barry will see the greatest impact from remote work by the year 2024, and Seattle by 2027.

Although the long-term ramifications of the COVID-19 pandemic on the workplace are still in limbo, modeling is nonetheless a powerful tool to make useful predictions. As the world settles upon a new normal, we believe that continuing to collect data and improving upon previous models will ultimately allow us to keep modeling the future.

Contents

Global Definitions

- We define a remote-ready job as a position where an employee can satisfactorily complete the position objectives without doing so from a workplace.

Global Assumptions

1. *There will be no significant policy changes regarding standards for remote work in the next 5 years.* Due to the unpredictability of new legislation, we cannot account for these changes.
2. *In accordance with our definition of "remote-ready," partially remote jobs do not qualify as remote-ready.* The complete lack of time in the workplace is inherent to being completely remote and thus "remote-ready."
3. *An adult is defined across both the US and the UK as a person 18 years of age or older, and a child is defined as under 18 years of age.* In order to differentiate between adults and children when analyzing census data, the distinction is imperative for simplicity's sake.

1 Part I: Ready or Not

1.1 Restatement of the Problem

We are tasked with creating a model to estimate the percentage of jobs currently ready for remote work and then use said model to predict the percentage of remote-ready jobs in 2024 and 2027. We will apply this model to the following cities:

- Seattle, WA, US,
- Omaha, NE, US,
- Scranton, PA, US,
- Liverpool, England, UK,
- Barry, Wales, UK.

1.2 Assumptions

1. *In accordance with our definition of "remote-ready," partially remote jobs do not qualify as remote-ready.* The complete lack of time in the workplace is inherent to being completely remote and thus "remote-ready."

2. *COVID-19 developments during or after 2023 will not affect a worker's status as remote-ready or not remote-ready.* Modeling by Emory University [3] and experts [11] predict that COVID will be endemic ("circulating in the general population") in the US and UK by 2023. Accordingly, new COVID-19 variants and infection spikes will not change a worker's status as remote or in-person.

3. *Public sentiment towards working remotely will not significantly change from the present.* Although firms may attempt to influence the public's perception of working remotely, the evidence regarding similar or greater productivity levels of remote workers compared to in-person workers [14] and the favorable anecdotal experiences of a large portion of the population regarding remote work during the COVID pandemic period [8] (2020-2022) will keep the public's perception of remote work as moderately favorable.

4. *We do not account for automation's future impact on the sector makeup of the labor force.* Automation varies significantly based on the urban versus rural characteristics of cities and towns. For example, Seattle, WA, a high-growth hub according to the McKinsey Institute, will be automated in 2027 to a higher degree than Scranton, PA, in the "mixed middle" of economic growth [7]. Additionally, the McKinsey Institute indicates that the bulk of automation will manifest on a 10-15 year timeline, rather than the 5-year timeline to 2027.

1.3 Variables Used

Symbol	**Definition**	**Units**
$\hat{P}_i$	Predicted number of people in industry i in a given year	People
μ_i	Percentage of individuals in industry i who can work from home	...
RR_c	Number of individuals in city c who are remote-ready	People

1.4 Model Development

For any city, the number of individuals who are remote-ready is given by

$$RR_c = \sum \hat{P}_i \cdot \mu_i,$$

where $\hat{P}_i$ is the predicted number of individuals in industry i in city c in a given year, and μ_i is the percentage of those individuals who are remote-ready. In order to predict the number of individuals in any city who are remote-ready, we must first take into account how many individuals are employed in each industry for that city during a given year, $\hat{P}_i$, since each industry has its own characteristics and responses to remote work pressures. In order to do this, we performed linear regression on the D1 city employment data provided by the 2022 MathWorks Math Modeling Challenge [6]. The number of individuals employed in each industry for each given city is the response variable, while the year, ranging from 2000 to 2021, is the explanatory variable.

The reason we chose linear regression was simply because of the large number of regression analyses needed to be computed. It is impractical to analyze each industry for each city and decide whether it follows an exponential, polynomial, or any type of pattern. Therefore, linear regression was employed because it offers a simple, consistent measure of predicting industry growth or decline in any given city.

It is worth noting that UK cities did not have data for the year 2000. The results of each linear regression are as follows:

Seattle

Industry	**2024**	**2027**
Mining, logging, constr.	122281	125811
Manufacturing	157608	152571
Trade, transp., and util.	378727	387780
Information	140115	149243
Financial activities	93710	92559
Professional and bus.	303718	316277
Education and health	275938	287018
Leisure and hospit.	170387	172776
Other services	72090	73726
Government	255364	256008

Omaha

Industry	**2024**	**2027**
Mining, logging, constr.	30905	32015
Manufacturing	32617	32452
Trade, transp., and util.	91253	89586
Information	9023	8314
Financial activities	46911	48322
Professional and bus.	75092	77036
Education and health	84683	88202
Leisure and hospit.	49255	50274
Other services	19165	19652
Government	68482	69854

Scranton

Industry	**2024**	**2027**
Mining, logging, constr.	10003	9946
Manufacturing	22881	20657
Trade, transp., and util.	64761	65854
Information	1730	1050
Financial activities	12727	12646
Professional and bus.	28168	28786
Education and health	53975	54826
Leisure and hospit.	20523	20449
Other services	7492	7170
Government	27913	27382

Liverpool

Industry	2024	2027
Mining, logging, constr.	150979	153097
Manufacturing	108080	113806
Trade, transp., and util.	160456	171263
Information	75737	78003
Financial activities	23212	23451
Professional and bus.	43108	43553
Education and health	21309	20326
Leisure and hospit.	67673	68047
Other services	76853	77403
Government	22463	21717

Barry

Industry	2024	2027
Mining, logging, constr.	4061	4076
Manufacturing	4785	4775
Trade, transp., and util.	1143	1130
Information	3754	3689
Financial activities	3740	3855
Professional and bus.	6945	7158
Education and health	10953	11151
Leisure and hospit.	10333	10333
Other services	3098	3108
Government	10953	11151

While some r^2 values and correlation coefficients are exceptionally low, these low correlations for an industry versus year effectively means that predicted values for $\hat{P}_i$ will be close to the calculated average value of the 5-6 data points analyzed by the linear model. That is an outcome that remains reasonable, and the $\hat{P}_i$ predictions of our linear regression models with r^2 values close to zero, should not be discounted.

Since each industry's work is remarkably different and certain industries are more capable of transitioning to remote work than others, the percent of individuals employed in each industry who can work remotely must be quantified and taken into account by the model.

The Remote Work data provided by the 2022 MathWorks Math Modeling Challenge contains this information, broken down into different industries from the industries listed in D1 industry employment data. Therefore, the following re-categorizations have been made:

R^2 Values	Seattle	Omaha	Scranton	Liverpool	Barry
Mining, logging, constr.	0.364	0.601	0.108	0.49	0.004
Manufacturing	0.387	0.109	0.745	0.719	0.001
Trade, transp., and util.	0.429	0.695	0.917	0.987	0.011
Information	0.856	0.917	0.991	0.867	0.27
Financial activities	0.228	0.949	0.246	0.074	0.279
Professional and bus.	0.809	0.835	0.458	0.074	0.279
Education and health	0.668	0.956	0.565	0.667	0.307
Leisure and hospit.	0.063	0.491	0.007	0.038	0
Other services	0.343	0.782	0.752	0.102	0.001
Government	0.005	0.776	0.784	0.342	0.307

D1	**D3 changes these headers**
Mining, logging, construction	Farming, fishing and forestry; Installation, maintenance and repair; Construction and extraction; Building and grounds cleaning and maintenance
Manufacturing	Production
Trade, transportation, and utilities	Transportation and material moving
Information	Computer and mathematical; Legal; Life, physical and social science; Architecture and engineering
Financial activities	Business and financial operations; Sales and related
Professional and business services	Management; Office and administrative
Education and health services	Education, training and library; Healthcare practitioners and technical; Healthcare support
Leisure and hospitality	Arts, design, entertainment, sports and media; Personal care and service
Other services	Community and social service
Government	Protective service

After averaging the estimated percentage of jobs that can be done at home by occupation category given by D3 for each industry category given by D1, we obtain the following values of μ_i for each industry:

Industry	μ_i
Mining, logging, construction	0.50%
Manufacturing	1.00%
Trade, transportation, and utilities	3.00%
Information	78.00%
Financial activities	58.00%
Professional and business services	76.00%
Education and health services	35.00%
Leisure and hospitality	51.00%
Other services	37.00%
Government	6.00%

Then, the RR_c for each city can be calculated as a function of year with the equation

$$RR_c = \sum \hat{P}_i \cdot \mu_i,$$

and the results are displayed in Figures 1–5.

1.5 Results

Using the population of each industry, $\hat{P}_i$, that we found in the first portion and the percentages of jobs that are remote-ready by each industry, μ_i, we can then calculate the percentage of remote-ready jobs in 2024 and 2027 using our formula. This gives us the following results:

City	**Predicted Percentage of remote-ready jobs in 2024**	**Predicted Percentage of remote-ready jobs in 2027**
Seattle	32.16%	32.55%
Omaha	31.63%	31.84%
Scranton	26.45%	26.60%
Liverpool	24.50%	24.18%
Barry	35.78%	35.82%

These results are reasonable, proving that even though some r^2 values of the linear regression models had low r^2 values, the combination of all the different linear regression models yields a robust prediction for the Predicted Percentage for remote-ready in 2024. For example, the decrease in predicted percentage of remote-ready in Liverpool makes sense, since it is the only city that had a large increase in employment in the manufacturing industry over the years 2005-2021.

Furthermore, we can compare this predicted remote-ready data for 2021 with the provided M3 Mathworks data in sheet D4. Seeing that in the months from March 2020 to September 2021, the percentage of US workers who worked from home exclusively had a maximum value of 54%, but quickly corrected to around 35-25%, we can infer that our predicted percentage of remote-ready for the US served as a sustainable percentage.

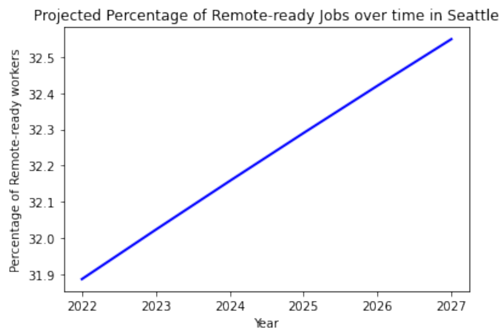


Figure 1: Seattle.

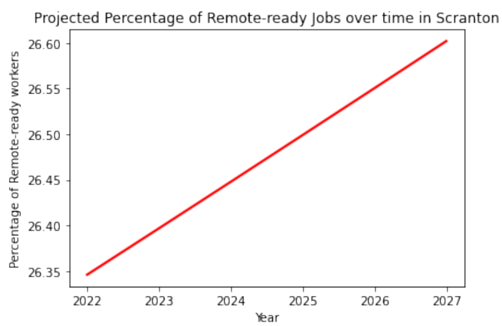


Figure 2: Scranton.

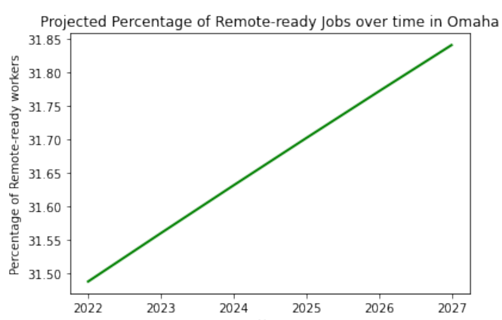


Figure 3: Omaha.

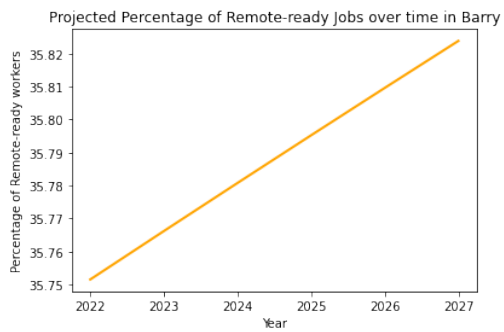


Figure 4: Barry.

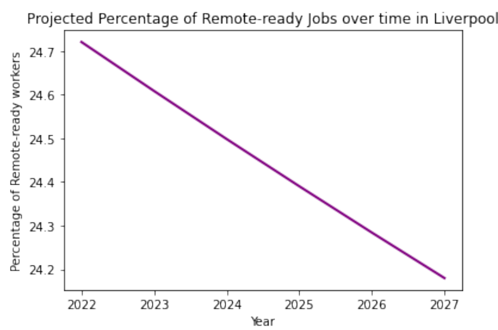


Figure 5: Liverpool.

1.6 Strengths and Weaknesses

A major challenge when considering the COVID-19 pandemic is its inherently volatile and unpredictable nature. It would be extremely difficult to account for all the possible scenarios, so it was necessary for us to make a simplification to create a useful model. Barring a major change like the emergence of a new variant, current models predict a trajectory in which the disease will become endemic in the US and UK by 2023. Assuming a constant endemic state for COVID-19 allows our model to not be affected by drastic changes in the disease's epidemiology between 2024 and 2027.

From the r^2 table, we can see a wide range of r^2 values. For example, most of the variability in the information industry employment in Scranton can be accounted for by the change in year ($r^2 = 0.991$). On the other hand, very little variation seen in the leisure and hospitality industry in Liverpool can be explained by the change in year ($r^2 = 0.007$). The extremely small data set contributes to this wide variability in the r^2 values. Our models could be improved by adding more data points, which would increase both the r and r^2 values to improve their prediction capabilities.

2 Part II: Remote Control

2.1 Restatement of the Problem

In this problem, we are tasked with creating a model to predict whether an individual worker with a remote-ready job will be allowed to and will choose to work from home.

2.2 Assumptions

1. *When considering an individual worker whose job is remote-ready, we assume that the probability that they are allowed to work from home by their employer and their desire to work from home are independent.* This allows us to individually consider each of these likelihoods and then find the product of them to calculate the probability of both events occurring. Additionally, Forbes reports a sizable disconnect between the groups on returning to the office [10].

2. *Employer sentiment towards remote work in 2021 will remain roughly constant until 2027.* While we understand that employer sentiment may vary by 2027, data on employer sentiment is only available up until the present. Additionally, we make the global assumption that COVID-19 will be endemic by 2024, so employer sentiment should not be largely affected by COVID-19 past that point.

3. *Only age, gender, and the presence of children in the household affect a worker's decision to work remotely.* This is the information our data included. Analysis from the Bureau of Labor Statistics supports that these are very important factors in the decision to work from home [13].

4. *All employers are equally likely to deny or allow fully remote work.* Data on variations by industry was scarce, and this assumption allowed us to focus on the random forest.

2.3 Variables Used

Age is an incredibly relevant factor in the decision to work remotely or in the workplace. The older members of the workforce may feel more comfortable with face-to-face interaction, whereas younger generations consider virtual interactions as commonplace and convenient. The Hartford reports 50% of small business owners ages 18-34 say remote workers are more productive than office workers, whereas only 15% of small business owners over 65 find remote workers more productive than in-person employees [12].

Gender is another indicator of the remote versus in-person work decision. A Flexjobs survey indicates 68% of women prefer to work remotely post-pandemic compared to only 57% of men. 80% of women consider it a key job benefit, whereas only 69% of men think the same [9]. This may be due to the higher proportion of housework that women do; the 2020 Women in the Workplace report found mothers were 1.5 times more likely than fathers to spend an extra three or more hours per day on housework [15]. Remote work is a method of balancing these requirements.

This leads into another factor to consider: parenthood. The additional duties of childcare can make remote work a helpful option to parents. Another Flexjobs survey found 61% of parents want to stay remote full-time, with 62% saying they would quit their current job if they cannot continue remotely [4].

On the other side of the decision to work remotely or in-person is the employer. As employees demand flexibility in work hours and arrangements, Owl Labs found 26% of employers are allowing employees to work remotely full-time [8]. We rounded this to 25% for ease.

2.4 Model Development

We employed the scikit-learn library in Python to train a random forest classifier model on the factors described above. Random forest classifiers employ decision trees trained on random samples of the data set to isolate variables and average the predictions of each tree, resulting in a robust prediction.

Our model is trained on the results of a random survey sourced from Kaggle regarding professionals' demographics and their decision to stay remote or return to the workplace [1]. Our target was the column "Same_office_home_location" (renamed "WFH"), or whether the professional's workplace was in the home. After splitting the data set into training and test sets, our model used 100 trees with a maximum depth of 5 nodes to make predictions regarding whether a given professional would choose to work from home.

To account for the approximate one-fourth of employers who are allowing employees to work from home full-time, we randomly generated a number from 1–4 inclusive using the random Python library in each instance that an employee's classification in the random forest was to work virtually. If the randomly generated number was 1, then the request was approved. This aligns with the approximate 25% of employers who are allowing fully remote work.

2.5 Results

The Random forest regression reached an **accuracy classification score of 0.74**. This is an acceptable accuracy given the time and data constraints present. We were additionally able to analyze the importance of the respective features using the feature_importances_variable from the random forest algorithm in scikit-learn.

We ultimately determined age is the most important feature with an importance of .75 in the classification model, followed by gender at .13 importance

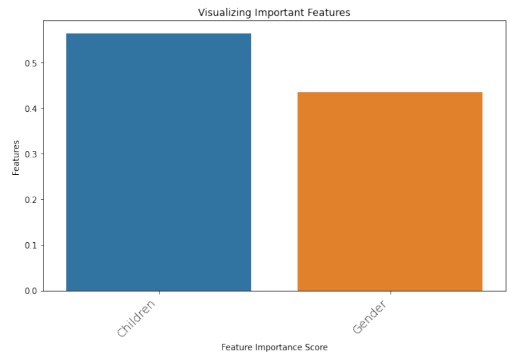


Figure 6: Important Features of Random Forest Classification.

and then children at .12 importance.

2.6 Strengths and Weaknesses

With a small data set of 207 respondents, the random forest classification's ability to mitigate the individual over fitting of decision trees with the averaging of a large number of decision trees is advantageous. This yielded a higher predictive accuracy than a single decision tree by itself.

Additionally, the random tree forest allows us to effectively combine features and identify the important features in the classification. We were able to determine that age is most closely linked to the employee decision to work remotely or in-person. However, this does bring us to a limitation of our data set, as we were not able to account for other factors likely to have an impact, such as education or income, because the data set did not include them. To improve upon this model, we would find a suitable data set with the factors of income and education and retrain the random forest. A specific shortcoming of the random forest model is the difficulty to ascertain the exact decision trees because of the large quantity of decision trees. We would also like to incorporate another random forest for the employer's choice, based on industry, company size, etc., given more time.

3 Part III: Just a Little Home-work

3.1 Restatement of the Problem

In this problem, we are tasked with creating a model that predicts whether an individual worker whose job is remote-ready will be allowed to and will choose to work from home.

3.2 Assumptions

1. *The working adult population is age 20-65.* The average retirement age is 62, so 65 is a reasonable upper cutoff [5]. Additionally, setting the lower cutoff at the age of 20 years allows for an easier analysis of census data, which groups ages by 5 years per stratum. Since these age bounds encapsulate the vast majority of working individuals, this was sufficient for our model.

2. *The demographics of the working adult population are consistent with the demographics of the general adult population of a given city.* This allows us to utilize census data, which is more readily available for the general adult population than for the working adult population.

3. *An individual's employment in a given industry is independent of their age, number of children, or gender.* While this is unlikely to be true, it allows us to conduct a simulation that determines the proportion of the remote-ready population that will actually work from home.

4. *The proportion of households with children that have 2 adults and the proportion of households with children that have 1 adult is consistent across the US and the UK.* If a household with children does not have 2 adults, we assume it has 1 adult. This allows us to find the average number of adults in a household with children.

5. *The age, gender, and child status data will not change over time.*

3.3 Variables Used

Symbol	**Definition**	**Units**
K_c	Proportion of adult population in city c that has children.	...
HC_c	Number of households in city c with children.	Households
F_c	Proportion of adult population of city c that is female.	...
P_c	Adult population of city c.	People
$A_c(Age_a - Age_b)$	Proportion of total adult population that has age between Age_a and Age_b for city c.	...

3.4 Model Development

The proportion of adults within a city with children, A_c, is given by

$$K_c = \frac{HC_c \cdot 0.69 \cdot 2 + HC_c \cdot 0.31}{P_c}.$$

Since a household that has children (people under 18), there is a probability of 69% that there are two parents in the household [2]. Since we have assumed that the remaining 31% of households with children have only one parent, the above equation yields the proportion of the adult population in a given city c that has children.

We used US and UK census data to find the number of households in each city that have children (HC_c), performed the necessary calculations, and have summarized the data in the following table:

$K_{Seattle}$	16.27%
K_{Omaha}	25.13%
$K_{Scranton}$	22.66%
$K_{Liverpool}$	21.54%
K_{Barry}	29.39%

Gender demographic data was similarly gathered from US and UK census data, and is shown below in the table for F_c:

$F_{Seattle}$	48.75%
F_{Omaha}	51.52%
$F_{Scranton}$	50.97%
$F_{Liverpool}$	50.58%
F_{Barry}	52.19%

To find the number of individuals in each age group for each city, we used the same census data for the US and UK. We performed the calculation

$$A_c(Age_a - Age_b) = \frac{I_c(Age_a - Age_b)}{WP_c},$$

where $I_c(Age_a - Age_b)$ is the individuals in each age range for each city and WP_c is the total working population (ages 20-65) for each city.

This will yield the $A_c(Age_a - Age_b)$, and we have summarized the data in the following tables:

We used a random number generator for values 0 to 1 for each of these 3 categorical variables. To account for differences in demographics for each city, we used K_c, F_c, $A_c(Age_a - Age_b)$ as weightings for the assignment of each categorical variable.

These individuals and their assigned categorical variables were passed into the random forest regression model from Q2 to predict whether each individual

City	Percentage that choose and employer choose GIVEN that they can
Seattle	8%
Omaha	6%
Scranton	9%
Liverpool	5.16%
Barry	6.81%

would choose to work from home, and that their employer would allow them to work from home. We multiplied this data by the predicted remote-ready percentages for the years 2021, 2024, and 2027 (which were calculated in Q1) to give the results of the simulation for the 5 cities for each of the 3 years.

3.5 Results

City	2021	2024	2027
Seattle	2.6296%	2.5728%	2.604%
Omaha	1.8672%	1.8978%	1.91%
Scranton	2.291%	2.38%	2.39%
Liverpool	1.31%	1.264%	1.248%
Barry	2.35%	2.437%	2.44%

In order to rank the 5 cities, we first defined the definition of "magnitude of impact" of remote work as the net magnitude change in the percentage of individuals working remotely in a given city. The net change from 2021 to 2024 and from 2024 to 2027 has been displayed for each city below:

City	Net change in percent from 2021 to 2024	Net change in percent from 2024 to 2027
Seattle	-0.056800%	0.031200%
Omaha	0.030600%	0.012600%
Scranton	0.089100%	0.013500%
Liverpool	-0.046440%	-0.016512%
Barry	0.091254%	0.002724%

3.6 Strengths and Weaknesses

From the results, it is clear that our simulation is underreporting magnitude of change in percent. We are unsure of why this happened, but it may have to do with our simulation weighting methods. Given more time, we could investigate this further.

Magnitude of change from 2021 to 2024 (ranked)	Magnitude of change from 2024 to 2027 (ranked)
Barry	Seattle
Scranton	Liverpool
Seattle	Scranton
Liverpool	Omaha
Omaha	Barry

We would also like to incorporate another random forest for the employer's choice, based on industry, company size, etc., given more time.

References

[1] A. Simon. "Predict if people prefer WFH vs WFO post Covid-19": Kaggle.

[2] U.S. Census Bureau. "The majority of children live with two parents, Census Bureau reports," Oct 2021.

[3] "Just another common cold virus? Modeling SARS-CoV-2's future fade": Emory University, Atlanta, GA, Jan 2021.

[4] "FlexJobs survey: Working parents want remote work": FlexJobs, Jan 2022.

[5] L. Konish. "Here is the age when many Americans hope to retire": CNBC, Jan 2022.

[6] 2022 MathWorks Math Modeling Challenge (M3 Challenge) Data, Feb 2022.

[7] "The future of work in America": McKinsey Institute.

[8] "State of remote work in 2021": Owl Labs.

[9] R. Pelta. "Survey: Men & women experience remote work differently": FlexJobs, Jan 2022.

[10] E. Segal. "The great disconnect: Many more employers than workers want to return to offices": Forbes, Dec 2021.

[11] S. Kimball. "Moderna says Covid is entering an endemic phase, but annual vaccines will be needed": CNBC, Feb 2022.

[12] The Hartford and K. Spors. "Younger generations leading the remote working charge": The Hartford, Dec 2020.

[13] H. Frazis. "Who telecommutes? Where is the time saved spent?": U.S. Bureau of Labor Statistics.

[14] "The survey of working arrangements and attitudes": WFH Research.

[15] "Women in the workplace": McKinsey & Company.

4 Appendix

Question 1 Model: Linear Regression

In [482…

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
```

In [483…

```
#Employment data for each city

def indexing(df):
    df.index = df['Industry']
    return df.drop(['Industry'], axis=1)

#Seattle, Washington
WAD1 = indexing(pd.read_csv ('SeattleWA.csv'))

#Omaha, Nebraska
NED1 = indexing(pd.read_csv ('OmahaNE.csv'))

#Scranton, Pennsylvania
PAD1 = indexing(pd.read_csv ('ScrantonPA.csv'))

#Liverpool, England
ENGD1 = indexing(pd.read_csv ('LiverpoolENG.csv'))

#Barry, Wales
WALD1 = indexing(pd.read_csv ('BarryWAL.csv'))

# Percentage of workers in each industry that can work from home
remoteJobsPercentages = pd.DataFrame({'Percentages': [0.005, 0.01, 0.03, 0.78, 0.58, 0.76, 0.35, 0.51, 0.37, 0.06]}, index=['Mining, logging, construction', 'Manufacturing', 'T
```

In [484…

```
#Linear regression on an single city industry
#Separated from UKlinReg because the US cities have data for 2000 whereas the UK cities do not
def USlinReg(series, city):
    # Create x-values of years
    x = np.array([2000, 2005, 2010, 2015, 2019, 2020, 2021]).reshape((-1, 1))

    # Convert the inputted series into a numpy array
    y = series.to_numpy()

    # Create linear regression model and fit points
    model = LinearRegression()
    model.fit(x, y)

    # Predict the y values for 2022-2028
    pred_x = np.array([2022, 2023, 2024, 2025, 2026, 2027, 2028]).reshape((-1, 1))
    pred_y = model.predict(pred_x)

    # Get name of industry
    industry = (series.name).lower()

    # Return values needed as tuple (2024, 2027, r^2)
    return (pred_y, r2_score(y, pred_y))
```

In [485…

```
#Linear regression on an single city industry
#Separated from USlinReg because the US cities have data for 2000 whereas the UK cities do not
def UKlinReg(series, city):
    # Create x-values of years
    x = np.array([2005, 2010, 2015, 2019, 2020, 2021]).reshape((-1, 1))

    # Convert the series inputted into a numpy array
    y = series.to_numpy()

    # Create linear regression model and fit points
    model = LinearRegression()
    model.fit(x, y)

    # Predict the y values for 2022-2028
    pred_x = np.array([2022, 2023, 2024, 2025, 2026, 2027]).reshape((-1, 1))
    pred_y = model.predict(pred_x)

    # Get name of industry
    industry = (series.name).lower()

    # Return values needed as tuple (2024, 2027, r^2)
    return (pred_y, r2_score(y, pred_y))
```

In [486…

```
#Performs linear regression for each industry of the inputted cityc
#Separated from UKcityRegression because the UK cities lack data for 2000 that the US cities have so the arrays are larger
def UScityRegression(df, city):
    #Creates arrays to store linear regression output of jobs
    totalJobs = np.array([0, 0, 0, 0, 0, 0, 0])
    totalRemoteJobs = np.array([0, 0, 0, 0, 0, 0, 0])
    #Iterates through each industry per year:
    for index, row in df.iterrows():
        amountOfJobs = USlinReg(row, city)[0]
        #Remote-ready jobs equals the percent of jobs per industry that are able to be done at home
        remoteReadyJobs = amountOfJobs * remoteJobsPercentages.loc[index]['Percentages']
        #Adds totals for each industry together for total jobs and remote-ready jobs
        totalJobs = np.add(totalJobs, amountOfJobs)
        totalRemoteJobs = np.add(totalRemoteJobs, remoteReadyJobs)

    #Computes percentage of remote-ready jobs out of total jobs
    percentageOfRemoteReady = (np.divide(totalRemoteJobs, totalJobs))*100
    #Returns percentage
    return percentageOfRemoteReady
```

In [487…

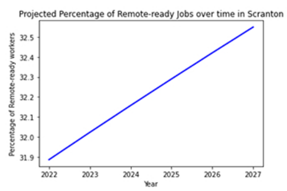
```
#Performs linear regression for each industry of the inputted city
#Separated from UScityRegression because the UK cities lack data for 2000 so the arrays are smaller
def UKcityRegression(df, city):
    #Creates arrays to store linear regression output of jobs
    totalJobs = np.array([0, 0, 0, 0, 0, 0])
    totalRemoteJobs = np.array([0, 0, 0, 0, 0, 0])
    #Iterates through each industry per year:
    for index, row in df.iterrows():
        amountOfJobs = UKlinReg(row, city)[0]
        #Remote-ready jobs equals the percent of jobs per industry that are able to be done at home
        remoteReadyJobs = amountOfJobs * remoteJobsPercentages.loc[index]['Percentages']
        #Adds totals for each industry together for total jobs and remote-ready jobs
        totalJobs = np.add(totalJobs, amountOfJobs)
        totalRemoteJobs = np.add(totalRemoteJobs, remoteReadyJobs)

    #Computes percentage of remote-ready jobs out of total jobs
    percentageOfRemoteReady = (np.divide(totalRemoteJobs, totalJobs))*100
    #Returns percentage
    return percentageOfRemoteReady
```

In [488…

```
#Creates graph for Seattle
seattle = UScityRegression(WAD1, "Seattle")[:-1]
x = np.array([2022, 2023, 2024, 2025, 2026, 2027]).reshape((-1, 1))
plt.xlabel("Year")
plt.ylabel("Percentage of Remote-ready workers")
plt.title("Projected Percentage of Remote-ready Jobs over time in Scranton")
plt.plot(x, seattle, color="blue", linewidth=2)
```

Out[488…

```
[<matplotlib.lines.Line2D at 0x7fdb47b25690>]
```

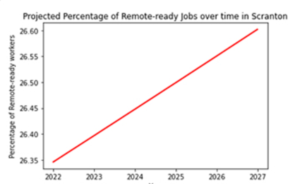


In [489…

```
#Creates graph for Scranton
scranton = UScityRegression(PAD1, "Scranton")[:-1]
x = np.array([2022, 2023, 2024, 2025, 2026, 2027]).reshape((-1, 1))
plt.xlabel("Year")
plt.ylabel("Percentage of Remote-ready workers")
plt.title("Projected Percentage of Remote-ready Jobs over time in Scranton")
plt.plot(x, scranton, color="red", linewidth=2)
```

Out[489…

```
[<matplotlib.lines.Line2D at 0x7fdb47aa3d10>]
```

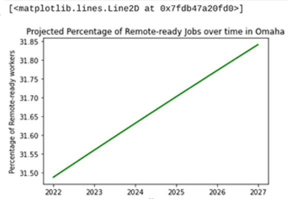


In [490…

```
#Creates graph for Omaha
omaha = UScityRegression(NED1, "Omaha")[:-1]
x = np.array([2022, 2023, 2024, 2025, 2026, 2027]).reshape((-1, 1))
plt.xlabel("Year")
plt.ylabel("Percentage of Remote-ready workers")
plt.title("Projected Percentage of Remote-ready Jobs over time in Omaha")
plt.plot(x, omaha, color="green", linewidth=2)
```

Out[490…

```
[<matplotlib.lines.Line2D at 0x7fdb47a20fd0>]
```

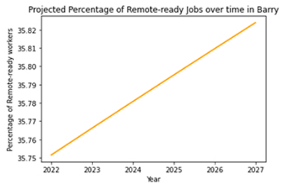


In [491…

```
#Creates graph for Barry
barry = UKcityRegression(WALD1, "Barry")
x = np.array([2022, 2023, 2024, 2025, 2026, 2027]).reshape((-1, 1))
plt.xlabel("Year")
plt.ylabel("Percentage of Remote-ready workers")
plt.title("Projected Percentage of Remote-ready Jobs over time in Barry")
plt.plot(x, barry, color="orange", linewidth=2)
```

Out[491…

```
[<matplotlib.lines.Line2D at 0x7fdb479b8110>]
```

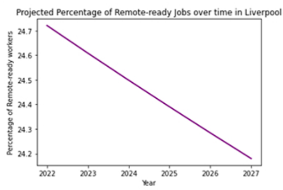


In [492…

```
#Creates graph for Liverpool
liverpool = UKcityRegression(ENGD1, "Liverpool")

x = np.array([2022, 2023, 2024, 2025, 2026, 2027]).reshape((-1, 1))
plt.xlabel("Year")
plt.ylabel("Percentage of Remote-ready workers")
plt.title("Projected Percentage of Remote-ready Jobs over time in Liverpool")
plt.plot(x, liverpool, color="purple", linewidth=2)
```

Out[492…

```
[<matplotlib.lines.Line2D at 0x7fdb47941150>]
```



In [ ]:

Question 2 Model - Random Forest

In [170…
```
#Import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import pandas_profiling

import os
import joblib
from sklearn.datasets import load_iris

from matplotlib import rcParams
import warnings

warnings.filterwarnings("ignore")
import random
#Figure size in inches
rcParams["figure.figsize"] = 10, 6
np.random.seed(42)
```

In [171…
```
#Load dataset
df = pd.read_csv("WFH_WFO_dataset.csv")

#Rename target variable to simplify
df['WFH'] = df['Same_ofiice_home_location']
df = df.drop('Same_ofiice_home_location', axis=1)

#Drop irrelevant columns
df = df.drop('Name', axis=1)
df = df.drop('Occupation', axis=1)
df = df.drop('ID', axis = 1)

#Replace string values with integers
df.loc[df['Gender'] == 'Female', 'Gender'] = 1
df.loc[df['Gender'] == 'Male', 'Gender'] = 0

df.loc[df['kids'] == 'Yes', 'Children'] = 1
df.loc[df['kids'] == 'No', 'Children'] = 0

#Drop redundant column
df = df.drop('kids', axis = 1)

#Print
print(df)
```

```
     Age Gender  WFH  Children
0     45      1  Yes       1.0
1     24      0   No       0.0
2     53      1  Yes       1.0
3     26      1  Yes       0.0
4     26      0  Yes       0.0
..   ...    ...  ...       ...
202   32      1  Yes       1.0
203   52      0  Yes       1.0
204   22      0  Yes       0.0
205   25      1   No       1.0
206   23      1   No       0.0

[207 rows x 4 columns]
```

In [172…
```
#Split data into input and target variable(s)
X = df.drop("WFH", axis=1) #input
y = df["WFH"] #target
```

In [173…
```
#Standardize the dataset
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [174…
```
#Split into train and test set
X_train, x_test, y_train, y_test = train_test_split(X_scaled,y,test_size=0.33, random_state=42)
```

In [175…
```
#Create the classifier
classifier = RandomForestClassifier(n_estimators=100)

#Train the model using the training sets
classifier.fit(X_train, y_train)
```

Out[175…
```
RandomForestClassifier()
```

In [176…
```
#Predict on test set
y_pred = classifier.predict(x_test)
```

In [177…
```
print(y_pred)
```

```
['No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No'
 'No' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'Yes'
 'Yes' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'Yes'
 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'Yes'
 'Yes' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes'
 'No' 'No' 'Yes' 'No']
```

In [178…
```
#Calculate Model Accuracy
print("Accuracy:", accuracy_score(y_test, y_pred))
```

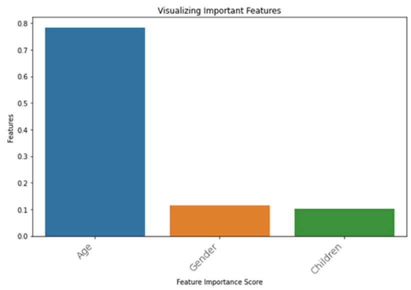
```
Accuracy: 0.7391304347826086
```

In [179…
```
#Check important features
feature_importances_df = pd.DataFrame(
    {"feature": list(X.columns), "importance": classifier.feature_importances_}
).sort_values("importance", ascending=False)

#Display
feature_importances_df
```

Out[179…

	feature	importance
0	Age	0.783094
1	Gender	0.114130
2	Children	0.102776

In [180…
```
#Visualize important featuers

#Create bar plot of important features
sns.barplot(x=feature_importances_df.feature, y=feature_importances_df.importance)

#Add labels
plt.xlabel("Feature Importance Score")
plt.ylabel("Features")
plt.title("Visualizing Important Features")
plt.xticks(
    rotation=45, horizontalalignment="right", fontweight="light", fontsize="x-large"
)
plt.show()
```



In [181…
```
joblib.dump(classifier, "./random_forest.joblib")
```

Out[181…
```
['./random_forest.joblib']
```

In [182…
```
loaded_rf = joblib.load("./random_forest.joblib")
```

In [193…
```
def runModel(csv):
    #Load dataset
    df = pd.read_csv(csv)

    #Rename target variable to simplify
    df['WFH'] = df['Same_ofiice_home_location']
    df = df.drop('Same_ofiice_home_location', axis=1)

    #Drop irrelevant columns
    df = df.drop('Name', axis=1)
    df = df.drop('Occupation', axis=1)
    df = df.drop('ID', axis = 1)

    #Replace string values with integers
    df.loc[df['Gender'] == 'Female', 'Gender'] = 1
    df.loc[df['Gender'] == 'Male', 'Gender'] = 0

    df.loc[df['kids'] == 'Yes', 'Children'] = 1
    df.loc[df['kids'] == 'No', 'Children'] = 0

    #Drop redundant column
    df = df.drop('kids', axis = 1)
    #Split data into input and target variable(s)
    X = df.drop("WFH", axis=1) #input
    y = df["WFH"] #target
    y_pred = loaded_rf.predict(X)
    #Accounting for employer choice (25%)
    for x in range(len(y_pred)):
        if y_pred[x] == 'Yes':
            #One in 4 employees will have their request approved by their employer.
            employer = random.randint(1, 4)
            if employer != 1:
                y_pred[x] = "No"
    return y_pred
```

In [194…
```
print(runModel('rftester.csv'))
```

```
['No']
```

Question 3 - Simulation

In [247…

```
#Import packages
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
import pandas_profiling

import os
import joblib
from sklearn.datasets import load_iris

from matplotlib import rcParams
import warnings

warnings.filterwarnings("ignore")
import random
```

In [248…

```
loaded_rf = joblib.load("./random_forest.joblib")
```

In [249…

```
def runModel(csv):
    #Load dataset
    df = pd.read_csv(csv)
    #Rename target variable to simplify
    df["WFH"] = ""

    #Replace string values with integers
    df.loc[df['Gender'] == 'Female', 'Gender'] = 1
    df.loc[df['Gender'] == 'Male', 'Gender'] = 0

    df.loc[df['kids'] == 1, 'Children'] = 1
    df.loc[df['kids'] == 0, 'Children'] = 0

    #Drop redundant column
    df = df.drop('kids', axis = 1)
    #Split data into input and target variable(s)
    X = df.drop("WFH", axis=1) #input
    y = df["WFH"] #target
    y_pred = loaded_rf.predict(X)
    #Accounting for employer choice (25%)
    for x in range(len(y_pred)):
        if y_pred[x] == 'Yes':
            #One in 4 employees will have their request approved by their employer.
            employer = random.randint(1, 4)
            if employer != 1:
                y_pred[x] = "No"
    return y_pred
```

In [250…

```
def simulationUSCity(age, gender, kids):
    df = pd.DataFrame(columns=['Age', 'Gender', 'kids'])
    for i in range(100):
        # Pick random numbers to find simulated age, gender, and if they have kids based upon cutoffs
        tage = random.random()
        if tage < age[0]:
            tage = 22
        elif age[0] <= tage < age[1]:
            tage = 29.5
        elif age[1] <= tage < age[2]:
            tage = 39.5
        elif age[2] <= tage < age[3]:
            tage = 49.5
        elif age[3] <= tage < age[4]:
            tage = 57.5
        else:
            tage = 62

        tgender = random.random()
        if tgender > gender:
            tgender = 0
        else:
            tgender = 1

        tkids = random.random()
        if tkids > kids:
            tkids = 0
        else:
            tkids = 1
        df.loc[len(df.index)] = [tage, tgender, tkids]
    df.to_csv('out.csv', index=False)
    result = runModel('out.csv')
    return result
```

In [251…

```
def simulationUKCity(age, gender, kids):
    df = pd.DataFrame(columns=['Age', 'Gender', 'kids'])
    for i in range(5000):
        # Pick random numbers to find simulated age, gender, and if they have kids based upon cutoffs
        tage = random.random()
        if tage < age[0]:
            tage = 22
        elif age[0] <= tage < age[1]:
            tage = 27.5
        elif age[1] <= tage < age[2]:
            tage = 37
        elif age[2] <= tage < age[3]:
            tage = 52
        else:
            tage = 62

        tgender = random.random()
        if tgender > gender:
            tgender = 0
        else:
            tgender = 1

        tkids = random.random()
        if tkids > kids:
            tkids = 0
        else:
            tkids = 1
        df.loc[len(df.index)] = [tage, tgender, tkids]
    df.to_csv('out.csv', index=False)
    result = runModel('out.csv')
    return result
```

In [252…

```
# Define our kid cutoffs by city
seattleKids = 0.1627
omahaKids = 0.2513
scrantonKids = 0.2266
liverpoolKids = 0.2154
barryKids = 0.2939

# Define our age cutoffs by city
seattleAge = [0.111063, 0.471448, 0.690171, 0.854297, 0.932333]
omahaAge = [0.118554, 0.387023, 0.606065, 0.801819, 0.898744]
scrantonAge = [0.157151, 0.386104, 0.606687, 0.794661, 0.895813]
liverpoolAge = [0.172402, 0.308475, 0.621761, 0.916863]
barryAge = [0.095833, 0.196734, 0.532044, 0.88901]

# Define our gender cutoffs by city
seattleGender = 0.4875
omahaGender = 0.5152
scrantonGender = 0.5097
liverpoolGender = 0.5058
barryGender =  0.5219
```

In [253…

```
# Seattle
seattleResult = simulationUSCity(seattleAge, seattleGender, seattleKids)
result = np.count_nonzero(seattleResult == 'Yes') / seattleResult.size
print(result * 100)
```

```
2.0
```

In [254…

```
# Omaha
omahaResult = simulationUSCity(omahaAge, omahaGender, omahaKids)
result = np.count_nonzero(omahaResult == 'Yes') / omahaResult.size
print(result * 100)
```

```
9.0
```

In [255…

```
# Scranton
scrantonResult = simulationUSCity(scrantonAge, scrantonGender, scrantonKids)
result = np.count_nonzero(scrantonResult == 'Yes') / scrantonResult.size
print(result * 100)
```

```
7.000000000000001
```

In [256…

```
# Liverpool
liverpoolResult = simulationUKCity(liverpoolAge, liverpoolGender, liverpoolKids)
result = np.count_nonzero(liverpoolResult == 'Yes') / liverpoolResult.size
print(result * 100)
```

```
5.46
```

In [257…

```
# Barry
barryResult = simulationUKCity(barryAge, barryGender, barryKids)
result = np.count_nonzero(barryResult == 'Yes') / barryResult.size
print(result * 100)
```

```
6.98
```