

Welcome to my portfolio

Below is a list of school and personal projects. Each has a description and links to a final report, source code, or a notebook.

Table of Contents:

- **Personal Projects**
 - **Data Collection and Cleaning**
 - *Scraping the IWF Website for Event Results And Athlete Data*
 - **Dashboards**
 - *Weightlifting Results Dashboad*
 - *Weightloss Tracking Dashboard*
 - **Machine Learning**
 - *Disaster Tweet Classification in R Using a linear SVM model*
 - *Predicting Survivors of the Titanic in Python*
 - *Recognizing Handwritten Digits with a Convolutional Neural Network using Python Deep Learning Frameworks*
 - **SQL and General Programming**
 - *Recreating Wordle in Python with a SQL Database for Scores*
- **Past School Projects**
 - **Visualization**
 - *Visualization and Exploration of the Gapminder Dataset Using ggplot/tidyverse Packages in R*
 - **Time Series Analysis**
 - *Analysis of Homicides in the US Over Time Using R and an ARMA/SARIMA Model*
 - **Regression Analysis**
 - *Reproducing the Results and Logistic Regression Model of a study on Modeling Prison Sentencing From Facial Features*
 - **Machine Learning**
 - *Predicting Ebay Car Prices Using a Random Forest Model in R*
 - *Fitting a Bayesian Hierarchical Model on Fake Flu Data. Simulated with an MCMC algorithm using R and Rjags/jags.*

Personal Projects

Data Collection and Cleaning

Scraping the IWF Website for Event Results And Athlete Data

The repository is github.com/cluffa/IWF_data. I scraped many pages for all event results and athlete stats from the International Weightlifting Federation's website. I used python for scraping. I needed the data to be easily imported into R, so I used R to clean and save the data.

I am currently exploring the data and working on an analysis. It will cover some topics such as competition strategy, comparison of countries and athletes, and predicting PED use.

Work in progress analysis [here](#).

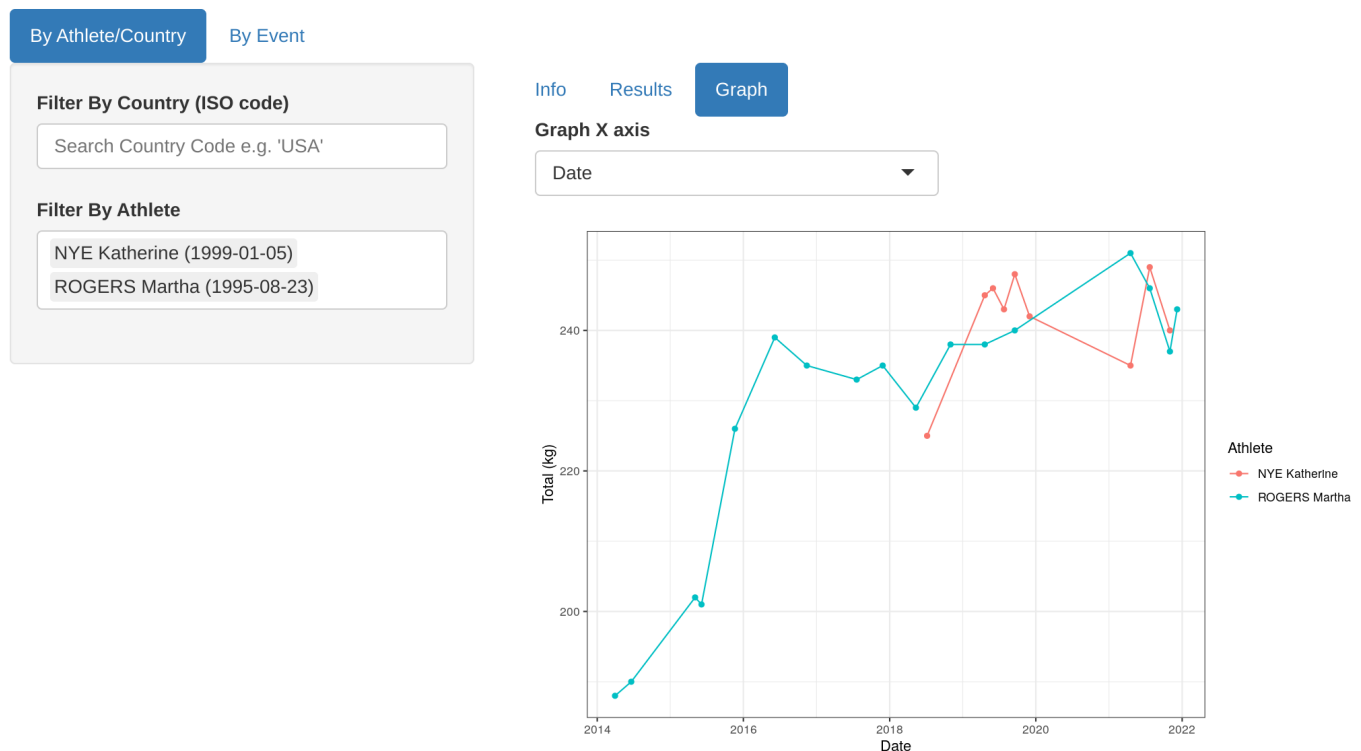
Dashboards

Dashboards are running [here](#) and the source code is at github.com/cluffa/MyShinyServer

I have a dashboard for easily filtering IWF event results and graphing athlete comparisons. There is also a dashboard I use for tracking weight loss trends with a linear regression model. It syncs with google sheets and my smart scale.

Weightlifting Results Dashboard

Explore IWF Event Results Data



Weightloss Tracking Dashboard

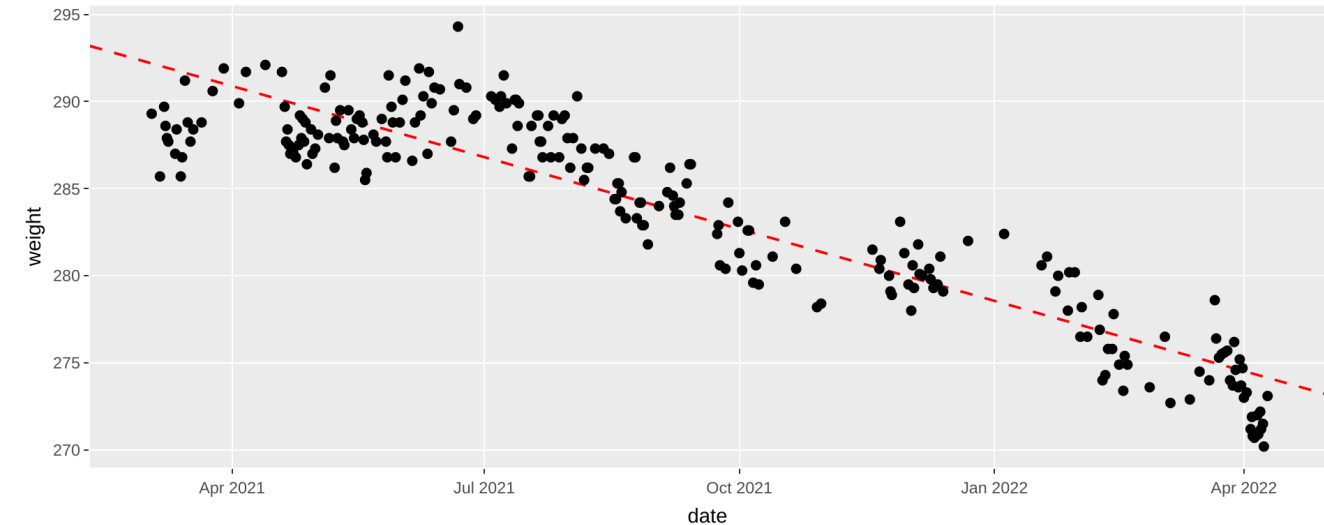
Weight Loss Trend

Date range input: yyyy-mm-dd

2021-03-01

to

2022-04-10



Daily Trend:
-0.04482101

Weekly Trend:
-0.3137471

Machine Learning

Disaster Tweet Classification in R Using a linear SVM model

View this project's [R Notebook](#).

The data is a collection of tweets that have been labeled as pertaining to a disaster or not. For example, one might be about the damage of an earthquake while another is about a sports team. Each tweet has a text body, keyword, and location. I used a linear support vector machine (SVM) model and tested the model with combinations of text body, keyword, and location.

Predicting Survivors of the Titanic in Python

View this project's [Jupyter Notebook](#). Most of the graphing and data exploration was done in the [first version of the notebook](#) where I added no features.

Id		Survived	Pclass		Name			
Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	
1	0	3				Braund, Mr. Owen Harris	male	
22.0	1	0	A/5 21171	7.2500	NaN	S		
2	1	1	Cumings, Mrs. John Bradley (Florence Brig...				female	
38.0	1	0	PC 17599	71.2833	C85	C		
3	1	3				Heikkinen, Miss. Laina	female	
26.0	0	0	STON/O2. 3...	7.9250	NaN	S		
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)				female	
35.0	1	0	113803	53.1000	C123	S		

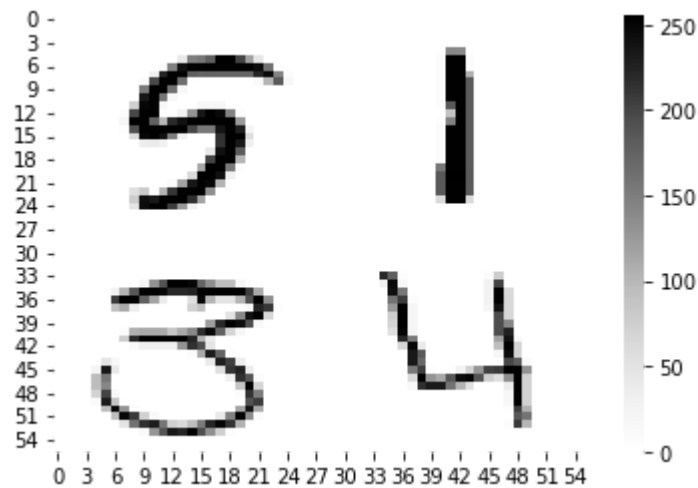
5	0	3	Allen, Mr. William Henry				male
35.0	0	0	373450	8.0500	NaN	S	

The main goal was to try to predict survivors based on what we know about each passenger. I used scikit-learn pipelines to make a clear transformation pipeline for the data. This includes encoding, multivariate imputing, as well as training. I used a gradient boosting classifier model where hyperparameters were optimized by grid search and cross-validation.

This notebook was used to submit scores to Kaggle's "Titanic: Machine Learning From Disaster" competition. With feature engineering like multivariate imputing and matching families, I achieved an accuracy score of 0.801 when submitting. This put me in the top 5% of the leader board.

Recognizing Handwritten Digits with a Convolutional Neural Network using Python Deep Learning Frameworks

View this project's [Jupyter Notebook](#) using neural networks and [Jupyter Notebook](#) using a standard machine learning model.



This is another Kaggle competition. The goal was to classify handwritten digits like the ones above. I wanted to compare the accuracy of traditional machine learning models with a convolutional neural network (CNN). I achieved 97.4% testing accuracy with an XGBoost model and 99.1% with a CNN using a TensorFlow keras sequential model. The traditional model did much better than I was expecting. However, these images are centered and scaled to be similar to each other. In a more uncontrolled environment I would expect the accuracy of the traditional type of model to drop off.


SQL and General Programming

Recreating Wordle in Python with a SQL Database for Scores

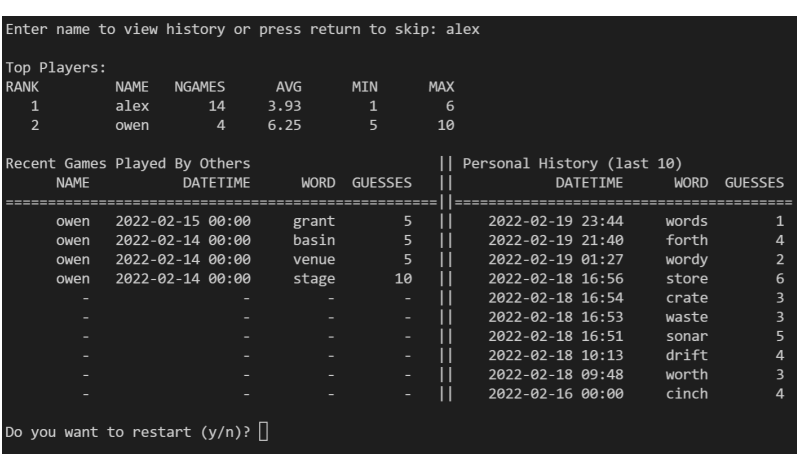
Repository: <https://github.com/cluffa/wordpy>

Game	Game History/Leader board
------	---------------------------

Game



Game History/Leader board



Some of the queries:

Logging

```
INSERT INTO
stats(
  NAME,
  date,
  word,
  guesses,
  uuid,
  words_guessed,
  can_lose,
  word_length,
  trys_lose,
  time_between
)
VALUES
(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s);
```

Rankings

```
SELECT
  rank() OVER (
    ORDER BY
      avg(guesses)
  ) rank,
  NAME,
  count(NAME) AS count,
  avg(guesses) AS avg_guesses,
  min(guesses) AS min_guesses,
  max(guesses) AS max_guesses
FROM
  stats
WHERE
  NAME != 'test'
GROUP BY
  NAME
ORDER BY
  avg_guesses;
```

Personal Game History

```
SELECT
  NAME,
  to_char(date, 'YYYY-MM-DD HH24:MI') AS fdate,
  word,
  guesses
FROM
  stats
WHERE
  NAME != 'test'
  AND NAME != %s
ORDER BY
  fdate DESC
LIMIT
  10;
```

This was a fun side project where my goal was to recreate wordle and use a SQL database for leader boards. I was able to further my understanding of databases and postgresSQL by using elephantSQL to host a table of the scores. Every time a game is completed, a log is added to the database with a name, date, word, number of guesses, and some other metrics. Then a query is executed to display basic stats and game history.

Past School Projects

Visualization

Visualization and Exploration of the Gapminder Dataset Using ggplot/tidyverse Packages in R

View this project's [final report](#) and [source code](#).

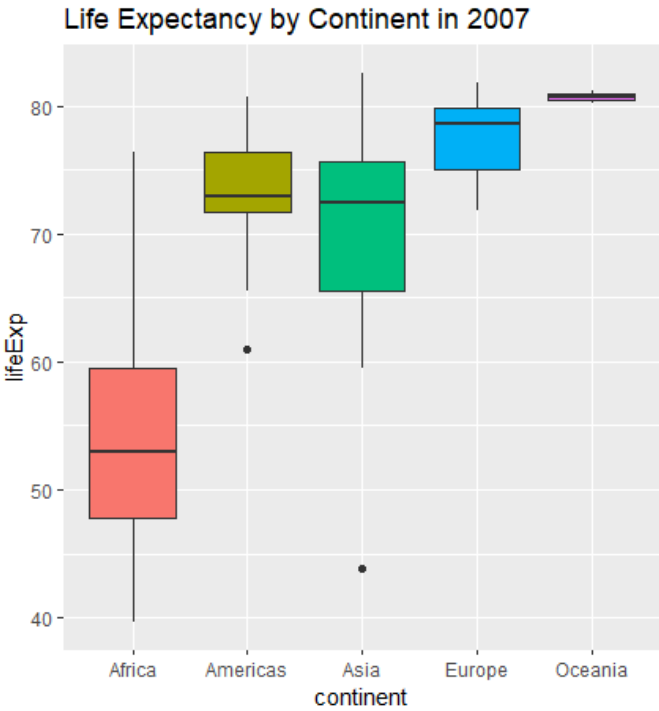


figure 3

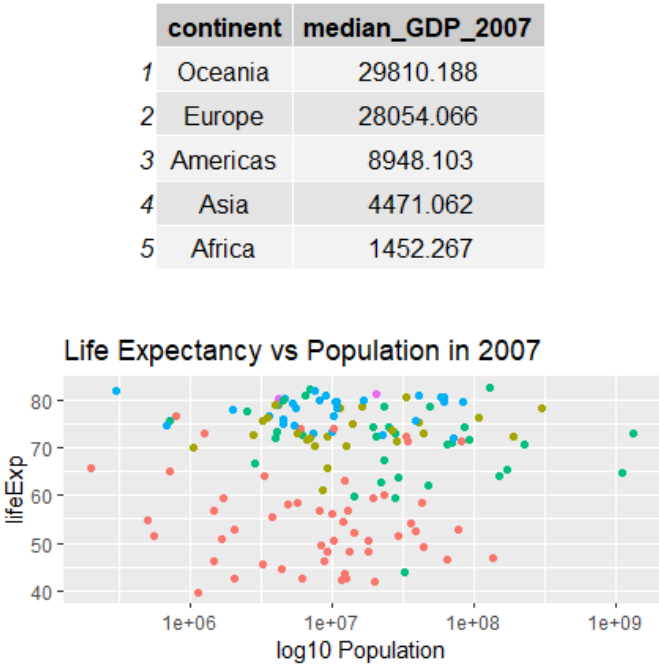


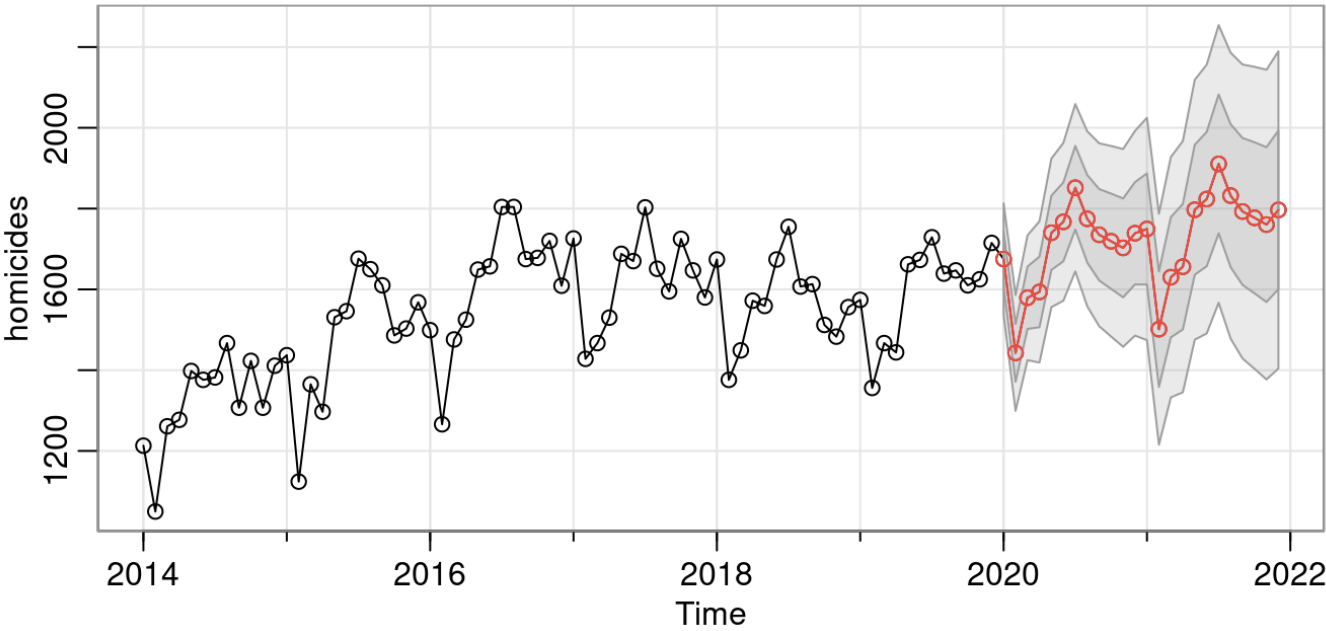
figure 4

The focus of this project was to explore the famous Gapminder dataset visually using ggplot graphs like the one above. I also used the other tidyverse packages like dplyr to manipulate the data in different ways to create well formatted data that fits into summary tables.

Time Series Analysis

Analysis of Homicides in the US Over Time Using R and an ARMA/SARIMA Model

View this project's [final report](#) and [source code](#).



This project was based around forecasting methods. I find overall trends, monthly seasonality, and fit ARIMA and SARIMA models. I compare the two model's performance as well as fit. I then forecasted homicides for the next 24 months.

Regression Analysis

Reproducing the Results and Logistic Regression Model of a study on Modeling Prison Sentencing From Facial Features

View this project's [final report](#) and [source code](#).

	Predictor	b	Pr(> z)	Std. Error	Odds Ratio	OR 95% CI
Model 1	Intercept	5.96	0.028	2.71	387.72	NA
	Trustworthiness	-1.55	0.022	0.68	0.21	[0.06, 0.8]
Model 2	Intercept	7.48	0.101	4.57	1780.52	NA
	Trustworthiness	-1.47	0.060	0.78	0.23	[0.05, 1.06]
	Afrocentricity	-0.51	0.213	0.41	0.60	[0.27, 1.34]
	Attractiveness	-0.30	0.725	0.86	0.74	[0.14, 3.98]
	Facial maturity	0.16	0.761	0.53	1.18	[0.42, 3.32]
	Presence of glasses	1.14	0.262	1.01	3.11	[0.43, 22.66]
	Time served	-0.14	0.067	0.08	0.87	[0.75, 1.01]

The idea for this assignment was to gain experience and become more comfortable reading and interpreting scientific research papers. We also learned the importance of reproducibility and transparency. My group was tasked with reproducing the results and models from this paper and reporting on them. I was in charge of the modeling as well as the table for the models, both of which are created with the source code I linked. The picture above is a replication of the table used in the original paper. Interestingly, we ended up finding a small mistake in the paper.

Machine Learning

Predicting Ebay Car Prices Using a Random Forest Model in R

View this project's [final report](#), [source code for the random forest model](#), and [source code for cleaning the data](#).

This was a group project. I handled the random forest model as well as the data cleaning. We each tried a model and compared results. The random forest model came out on top based on testing MSE. I had to learn a lot about resource allocation to complete this project. The dataset had 180k rows and 10+ possible predictors. I quickly found out that I would not be able to easily tune and train the model. The training process ended up taking a few days with cross-validation on less than half of the full dataset. I had to weigh run time vs accuracy and pick model parameters early on in the training process.

Fitting a Bayesian Hierarchical Model on Fake Flu Data. Simulated with an MCMC algorithm using R and Rjags/jags.

View this project's [final report](#), [report source code](#), and [model fitting source code](#)

The setup for this project:

"There are two tests for influenza strain K9C9. The data collected consists of 10 countries and 100 pairs of test results. The more accurate of the tests will be assumed fact. The less accurate test, EZK, is the area of interest for this project. A Bayesian hierarchical model will be fit and it will be simulated with an MCMC

algorithm using R/jags."

I fit the model, assessed fit, and interpreted the results in the context of a global pandemic.