Project 2

The goal of this project is to create a report using R Markdown that has code to pull data from an API, convert that data to an appropriate form, and fit models. This is an individual project and you should turn in your .Rmd file along with an HTML or PDF file.

API

Zillow API

For this project we're going to use the `ZillowR` package and the Zillow API. Zillow is a real estate web site that allows people to view homes that might be for sale (or aren't) and gives them things like an estimated price for the house.

You will need to obtain a Zillow API account and account key (`zws_id`). You can go to this web site to register.

Once you've installed the package, loaded it in, and obtained an id you should be able to modify and run the code below.

```
GetDeepSearchResults(address = '14707 W SUNNY DR', citystatezip = "Los Angeles, CA",
                                          zws_id = "your-key-here")
```

The output is a list and one of the list elements (`response`) has the data of interest in XML format. You will need to parse this data (see below for instructions). The variables you should grab from the output are

- Street number and name
- Zipcode
- City
- use Code
- tax assessment year
- tax assessment value
- year built
- lot size
- finished square feet
- bathrooms
- bedrooms
- zestimate amount (our response variable)
- region name
- region type

Not all of these will be available for every house. You should make sure you grab what is available and leave the other values as missing (`NA`).

List of Addresses

There is a list of addresses for the city of Los Angeles available here. We will take a subset of these addresses and obtain the data from zillow's API. Not all of these addresses will work (either the address doesn't seem to exist or zillow just doesn't have info on the address) so some results will return NULL. Your function to parse the data should utilize this to deal with those addresses.

Report

Introduction section

You should have an introduction section that describes the purpose, methods, and general conclusions from your analysis.

Data

We'll attempt to run an analysis to predict the Zestimate (Zillow's estimated property cost). Randomly sample from (say using `sample()`) and split the LA address data set into two parts, a test set and a training set. From playing around with the data, about 20-30% of the addresses return data. As zillow seems to only allow about 5000 calls to the API a day, you should be able to get at least 1000 values. If there is a missing value for the Zestimate, remove that observation. Let's form a training set with 80% of your data and a test set with 20%.

You should write code that reads the full address data set in, randomly selects an appropriate number of addresses to use, queries the API, processes the data (throwing out values that returned `NULL` for `$response`), and outputs a single CSV file with relevant information (variables noted above). *The querying and formating of the data should be done using a custom function you write.* (I would do this outside of markdown so you don't try to run it everytime you knit your file. In the end you should put your code (with appropriate text throughout) into your markdown document you'll turn in, but I'd use `eval = FALSE` on relevant chunks of code.)

Modeling

Once you have your training data set, we are ready to fit some models.

You should fit two types of models to predict the Zestimate. One should be an ensemble model (bagged trees, random forests, or boosted trees) and one should be a linear regression or tree model. Feel free to use code similar to the notes or use the `caret` package.

You can use whichever variables (or functions of the variables - perhaps standardized values) to predict the Zestimate except

- Street number and name
- Zipcode
- City

For the variables you choose to use in your modeling, you can remove any observations with missing values for the predictors used. Remember that some variables may look numeric but they aren't!

After training your models using cross-validation, AIC, etc. you should then compare them on the test set.

Conclusions

You should use your models to write a conclusions section that discusses the implications of the model you would choose to predict the Zestimate. Be sure to think about real world considerations with the variables you included in your model.

Rubric for Grading

Item	Points	Notes
Introduction	10	Worth either 0, 3, 7, or 10
Address data read in, randomly chosen from	5	Worth either 0, 2, or 5
API query and processing	20	Worth either 0, 4, 8, . . . , or 20
Data split	5	Worth either 0, 2, or 5
Data preprocessing	10	Worth either 0, 3, 7, or 10
Ensemble model fit	15	Worth either 0, 5, 10, or 15
Linear regression or tree fit	15	Worth either 0, 5, 10 or 15
Test set prediction	5	Worth either 0, 2, or 5
Conclusions	15	Worth either 0, 5, 10, or 15

Notes on grading:

- For items worth say 0, 5, 10, or 15 points, we will generally move you down one level for each syntax, logical, or other error present in the code. The same holds true for missing a required item or lacking in a description.
- Although not explicity in the criteria above, points will be taken off for not following good programming practices (up to 30) and for not using appropriate markdown options/formatting (up to 20)