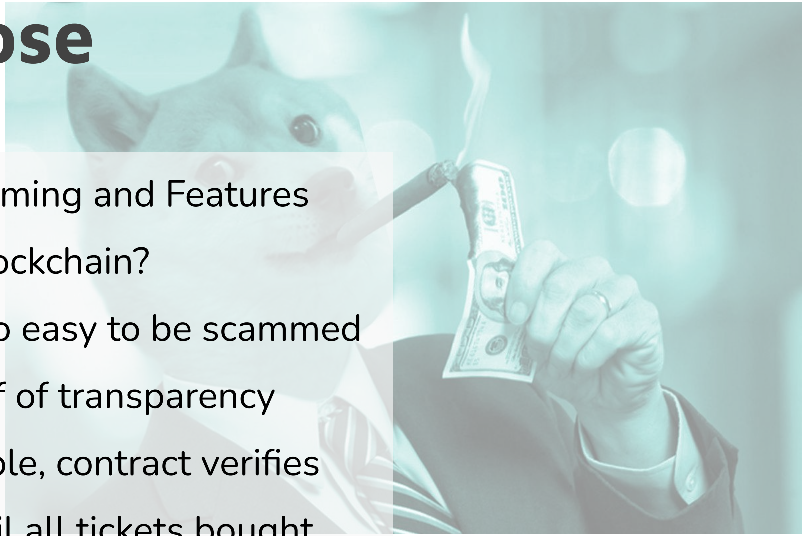
Blockchain Games of Chance

Alpha Gen Analytics

Introduction & Purpose



- Purpose : Explore Blockchain Programming and Features
- Can we find a pragmatic use of the Blockchain?
  - For online “games of chance”, too easy to be scammed
  - Blockchain offers verifiable proof of transparency
  - Lottery - All tickets publicly visible, contract verifies winning numbers not drawn until all tickets bought
  - Blackjack - Cards are drawn randomly, contract verifies the Dealer cannot cheat

Play the Lottery

- git clone https://github.com/BaldHeads/play_bChain_lotto
- Read how_to_play.md
- Use contract address 0x7c37ff45f8Df929bc6619324EF4077CDFC0a7368
- Use assigned private key in class Slack

Public Ganache Testnet

- Ganache-cli running on base Linode instance
- Connect to http://45.33.17.146:8545 with chain ID 5337 with MyCrypto, MetaMask, or Web3
- Initial account *was* seeded with 100 Billion ETH (but no longer)
- Python script to fund 2000 accounts with 100 ETH

Lottery Smart Contract Features

- Only Deployer can start and stop games
- Lotto rules - 6 tickets with a customizable number range number range from 1 to 255
- Charity option where a 2nd wallet gets 20% of winnings, or 100% if there are no winning tickets
- Emits Events for every ticket bought
- Random numbers generated in contract using keccak256 over block.timestamp and difficulty
- Contains a mapping of ticket ⇒ buyers so it’s easy to find winning buyers
- Cannot buy tickets after winning numbers are generated

Lottery Python UI Features

- Deployer .py file can create new games, connect to open games, list all tickets bought via Event Log, show lotto pot, and finalize lottery
- Player .py file connects to open lottery, can buy Pick 6, buy custom ticket, or show winning numbers of finished game

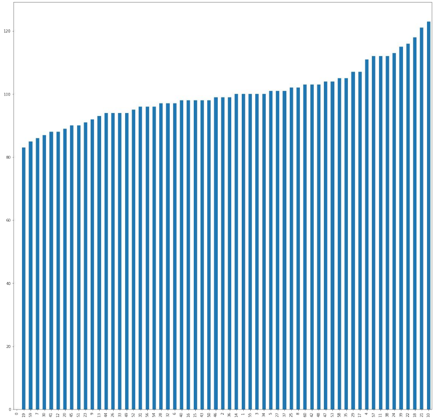
Lottery Number Analysis

Analyze past 1000 lotto drawings - Due to Ganache mining speed, each run on 1000 draws took 20 minutes

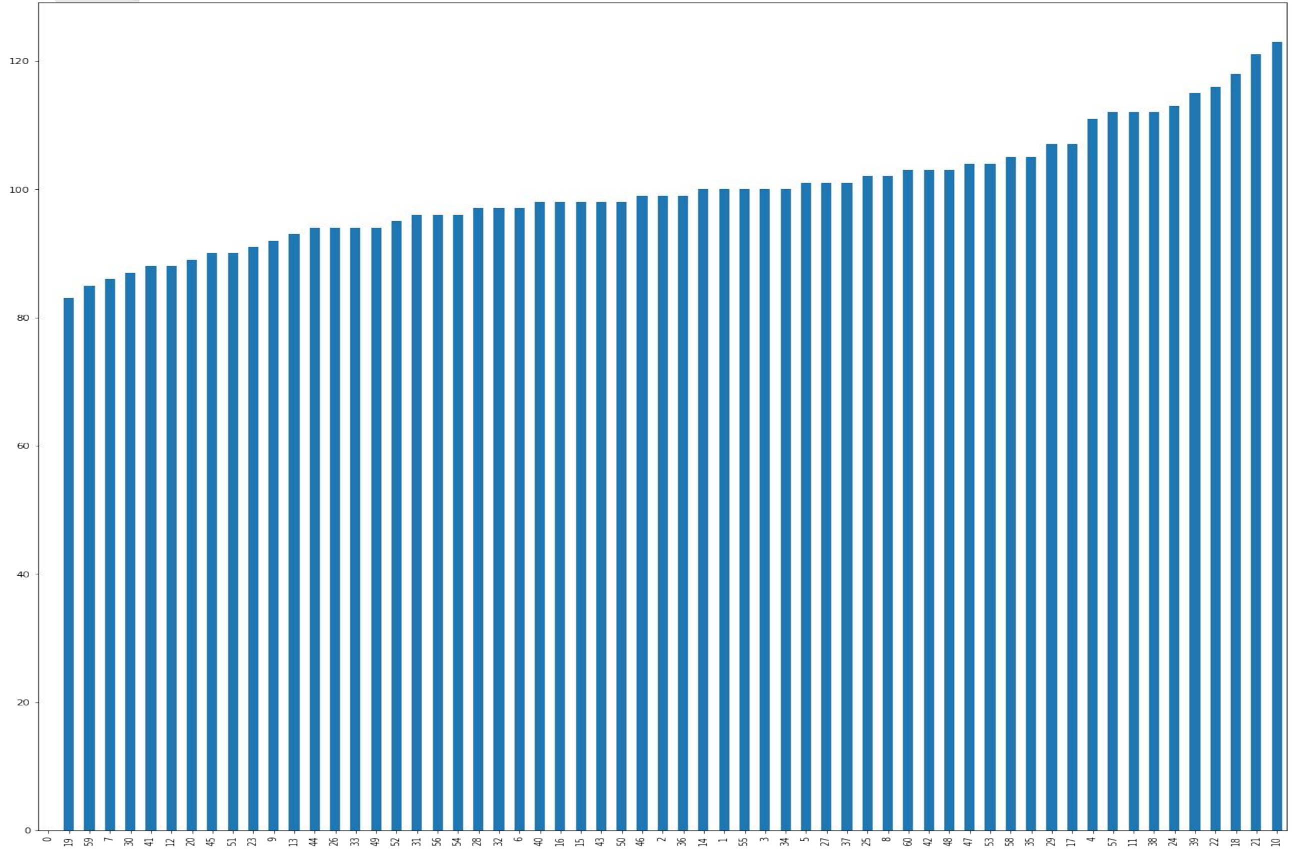
Run a frequency count on each individual number

- Chose 6 most frequent numbers (due to bad random number generator)
  - [10, 18, 21, 22, 24, 39]
- Chose 6 least frequent numbers (due to mean regression)
  - [7, 12, 19, 30, 41, 59]
- Predicted Tickets:
  - prediction1 = [7,11,21,35,42,60]
  - prediction2 = [10,19,21,28,43,56]
  - prediction3 = [2,9,20,34,41,44]
  - prediction4 = [1,17,33,35,46,52]

Run ARIMA model on each slot



A closer look at the Chart



Play BlackJack

- 2 options for with or without MetaMask
  - http://hardymachine.com.nightshade.arvixe.com/
  - http://hardymachine.com.nightshade.arvixe.com/js_metamask/
- Use assigned private key in class Slack

Blackjack Smart Contract Features

- Dealer fills pot before player ante's up
- Once player submits ante, game allows for hit, stand, double down, and buying insurance on dealer ace
- Dealer can end game anytime and return ante
- Contract stores both hands and hands are viewable at all times
- Deals Random cards and remembers dealt cards
- Dealer hand is generated on ante, but contract will only reveal face-up card until player busts or stands
- Pays out 1.5x on BlackJack
- Shuffle and Deal gas fees paid by player

Blackjack UI Features

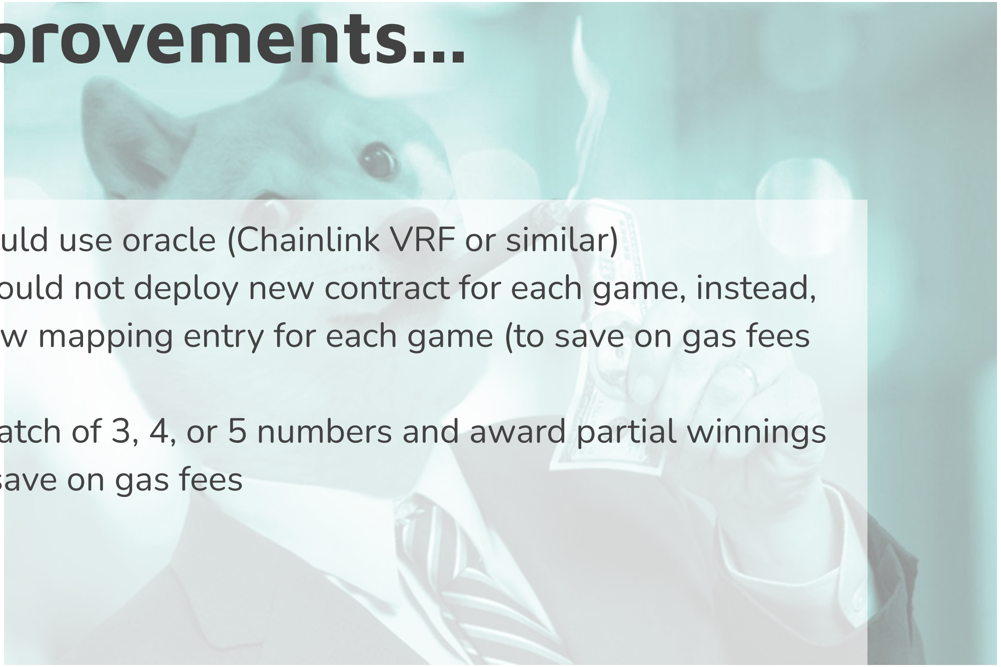
- Web dApp that connects to public testnet
- Shows player ETH balance before and after game
- Allows for hit, stand, and double down (not insurance)
- Option to use MetaMask or private key directly
- Uses Unicode fonts for card visualization

Outside Library

- Web3js - to work with blockchain on the web

- Ganache-cli : to run ganache on a public server over SSH
    - Note - do not store private keys in GitHub

V2 Planned Improvements...



- Random Number Generator should use oracle (Chainlink VRF or similar)
- Blackjack and Lottery games should not deploy new contract for each game, instead, single contract should hold a new mapping entry for each game (to save on gas fees for deploy)
- Lotto should allow for partial match of 3, 4, or 5 numbers and award partial winnings - would need custom oracle to save on gas fees
- Multiplayer Blackjack
- Poker

Final Words