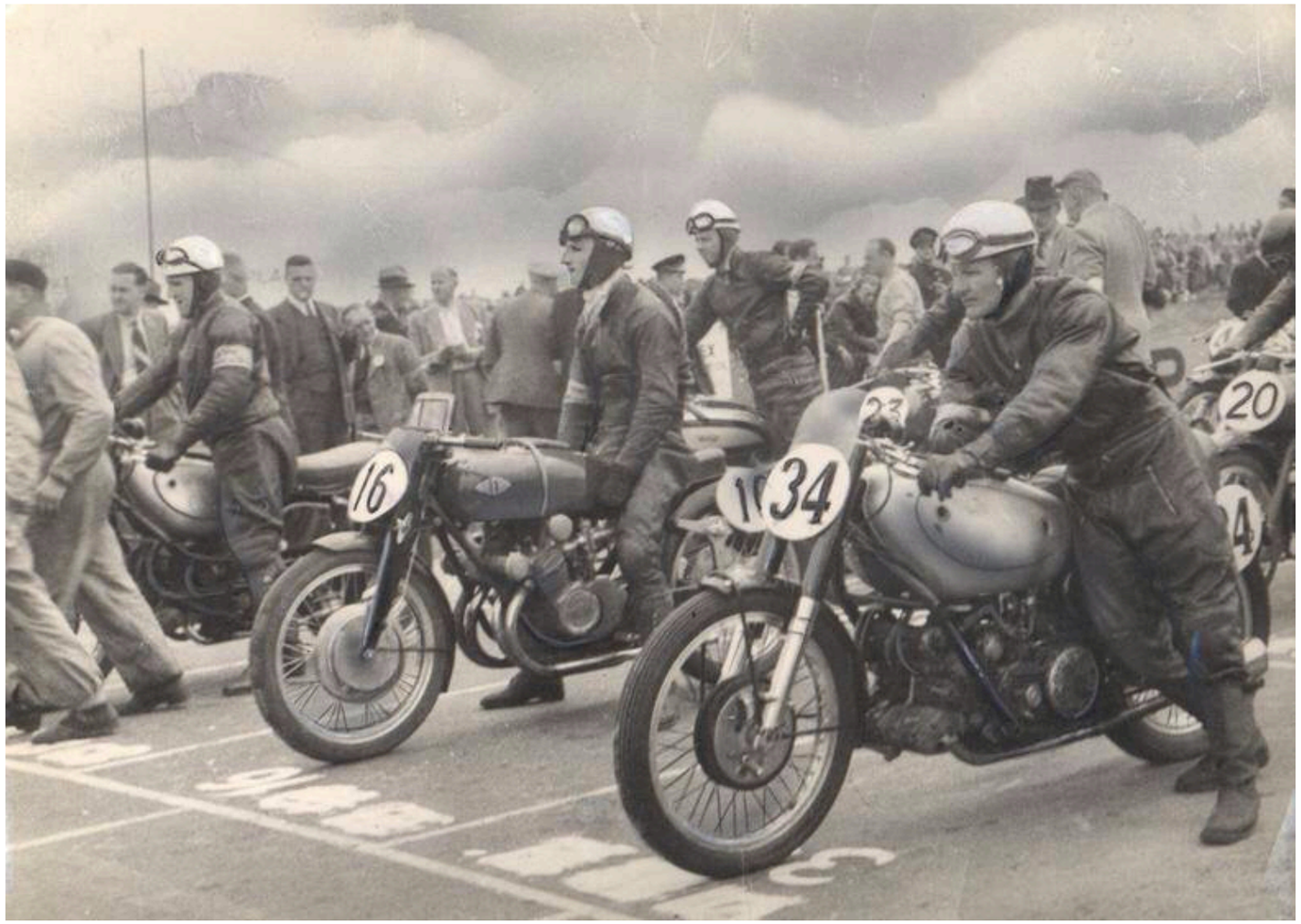
Predicting MotoGP race finish times using linear regression

Ankur Vishwakarma
Metis SF Winter 2018

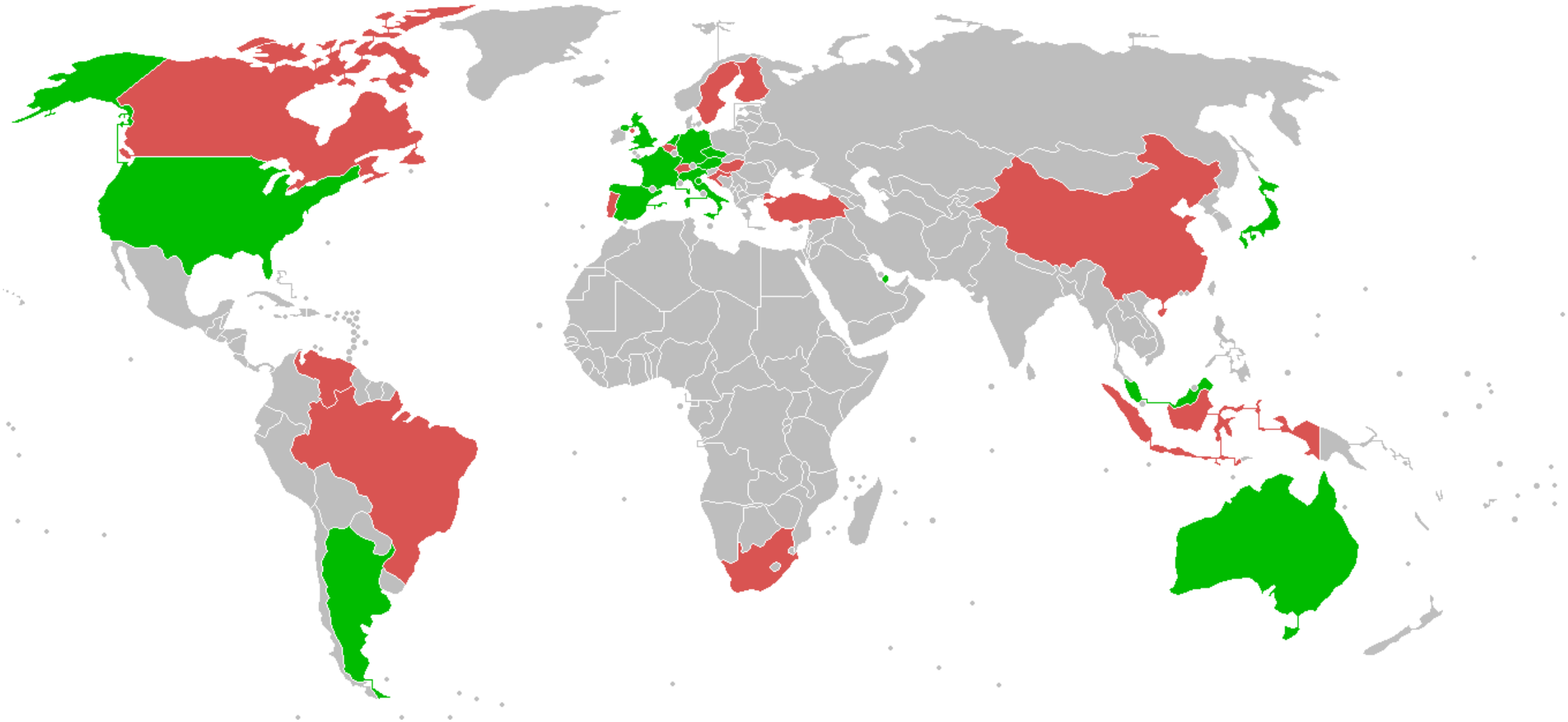
MotoGP 101

- Motorcycle Grand Prix racing
- Started in 1949



MotoGP 101

- International racing series



MotoGP 101

- 3 classes
  - **Moto3**
    - Entry series
    - Last year's champion was 20 years old
    - Top speeds of 152 mph
  - **Moto2**
    - Intermediate series
    - 174 mph
  - **MotoGP**
    - Premiere class
    - Prototype machines
    - Over 220 mph

MotoGP 101

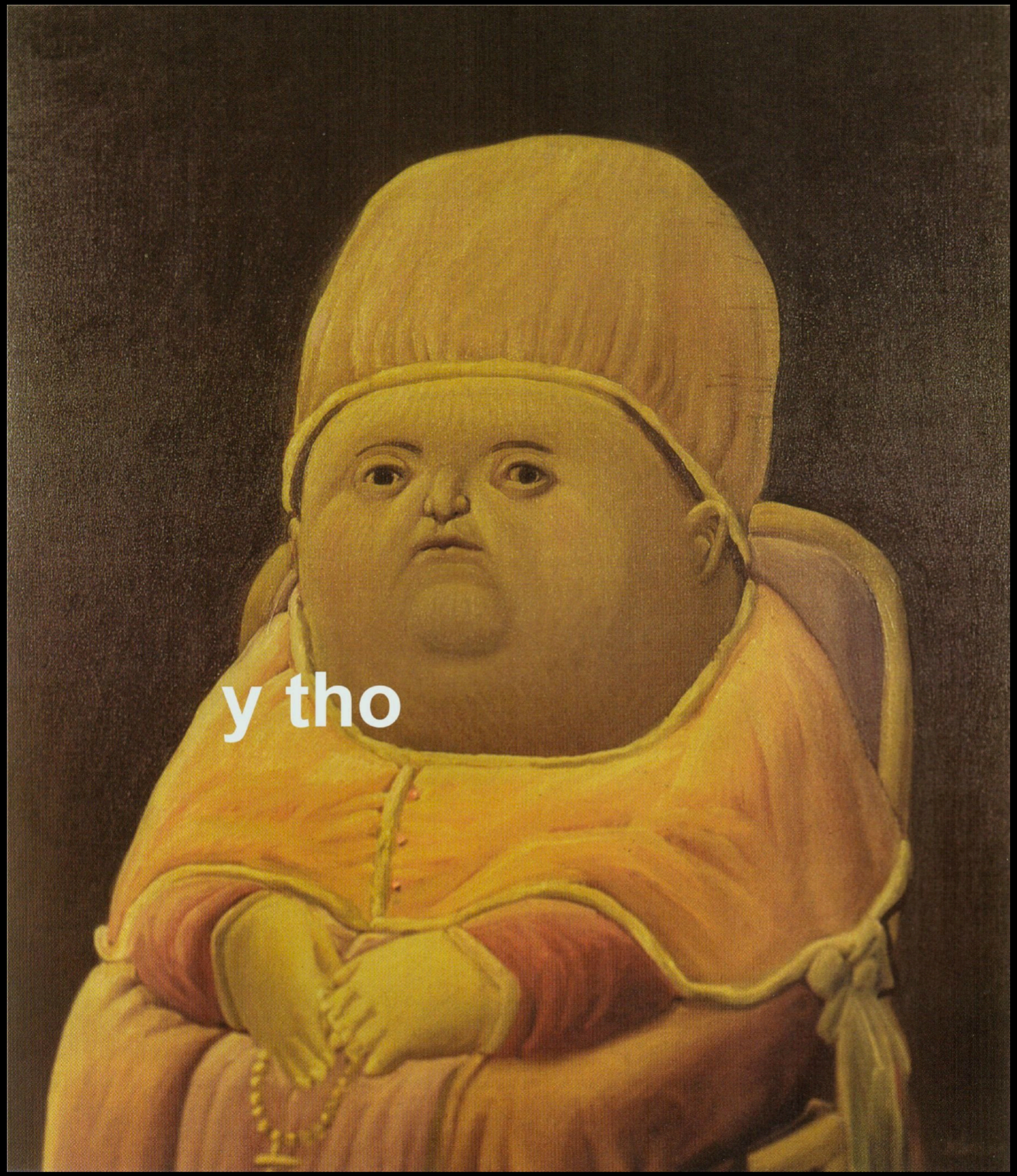
- Points are awarded for finishing position at each race.
- Rider with the most points at the end of the season becomes the world champion.

Pos.	Points	Num.	Rider	Nation	Team	Bike	Km/h	Time/Gap
1	25	46	Valentino ROSSI	ITA	Movistar Yamaha MotoGP	Yamaha	169.9	41'41.149
2	20	9	Danilo PETRUCCI	ITA	OCTO Pramac Racing	Ducati	169.9	+0.063
3	16	93	Marc MARQUEZ	SPA	Repsol Honda Team	Honda	169.6	+5.201
4	13	35	Cal CRUTCHLOW	GBR	LCR Honda	Honda	169.6	+5.243
5	11	4	Andrea DOVIZIOSO	ITA	Ducati Team	Ducati	169.6	+5.327
6	10	43	Jack MILLER	AUS	EG 0,0 Marc VDS	Honda	168.3	+23.390
7	9	17	Karel ABRAHAM	CZE	Pull&Bear Aspar Team	Ducati	167.4	+36.982

Question

- Can we predict **time to finish a race** based on:

	Type	Variables
1	Track characteristics	Length, longest straight, number of corners, width, past average speeds, etc.
2	Weather data	Air temperature, track temperature, humidity
3	Rider’s skill	Mean of points/race throughout career
4	Motorcycle characteristics	Class (MotoGP vs Moto3)


y tho

Why?

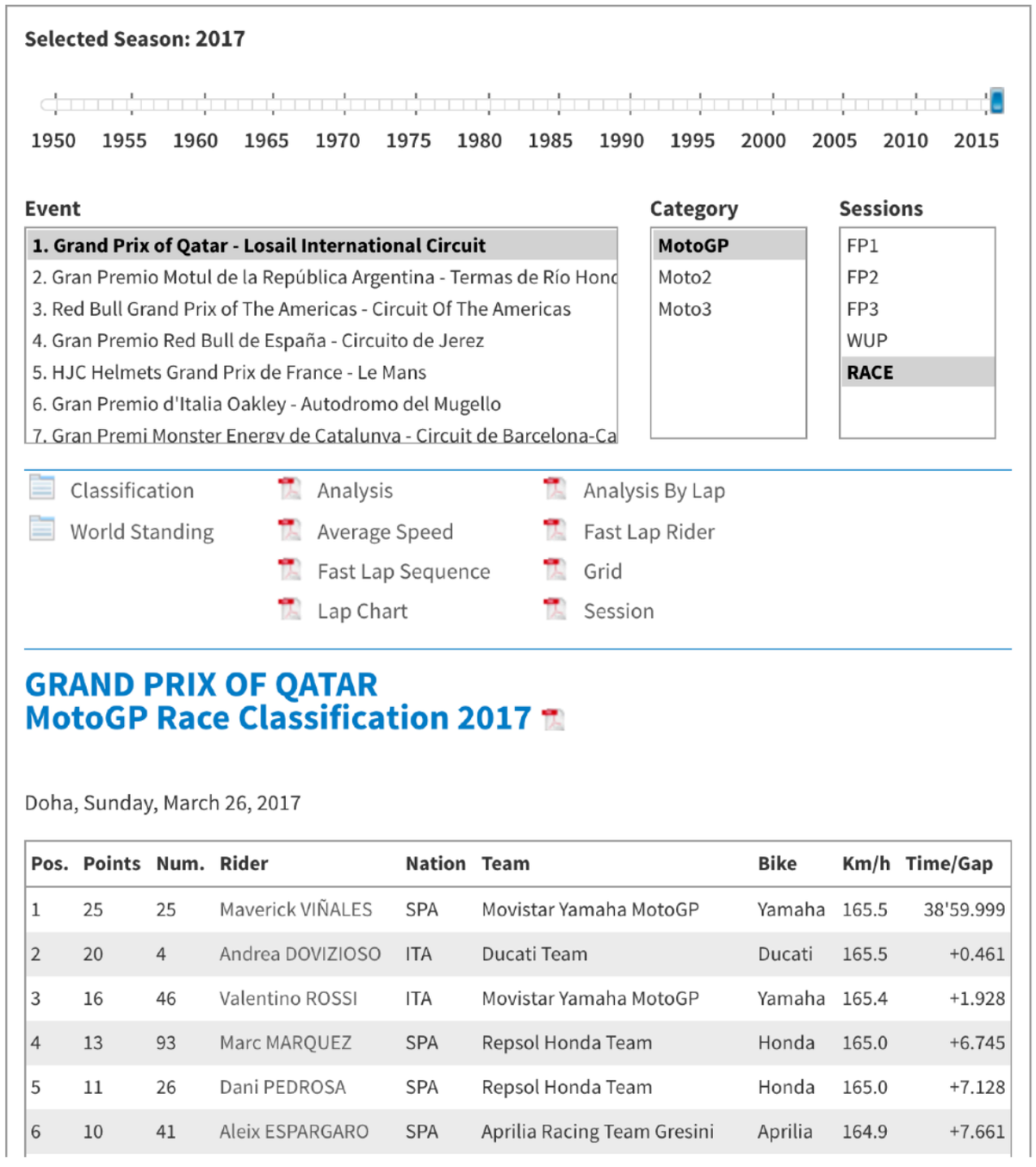
- Imagine you're a MotoGP team director:
  - Recruit a new rider from the junior classes.
  - Predict their performance numerically.

- Ability to predict time for riders for new racetracks.

- Why predict the time and not the position?
  - Because finishing time is continuous.
  - Position 2.33 ± 0.57 doesn't make a lot of sense.

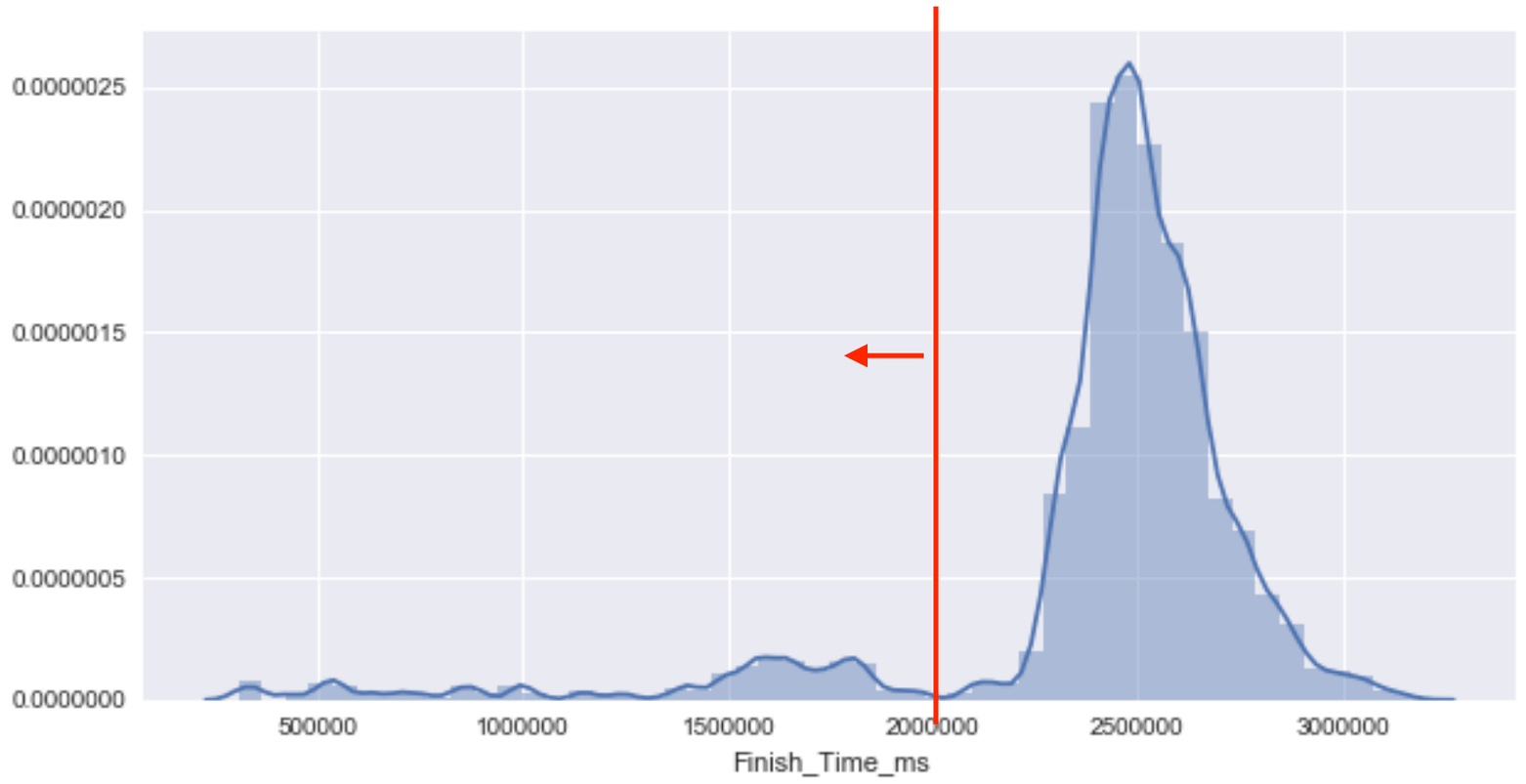
Step 0 - Scraping

- 2005 ⟶ 2017
- BeautifulSoup
- Load HTML
  - Parse <options>, <a href>, <tr>, etc.
- Scrape, Rinse, Repeat
- 19,412 x 30



Step 1 - Cleaning

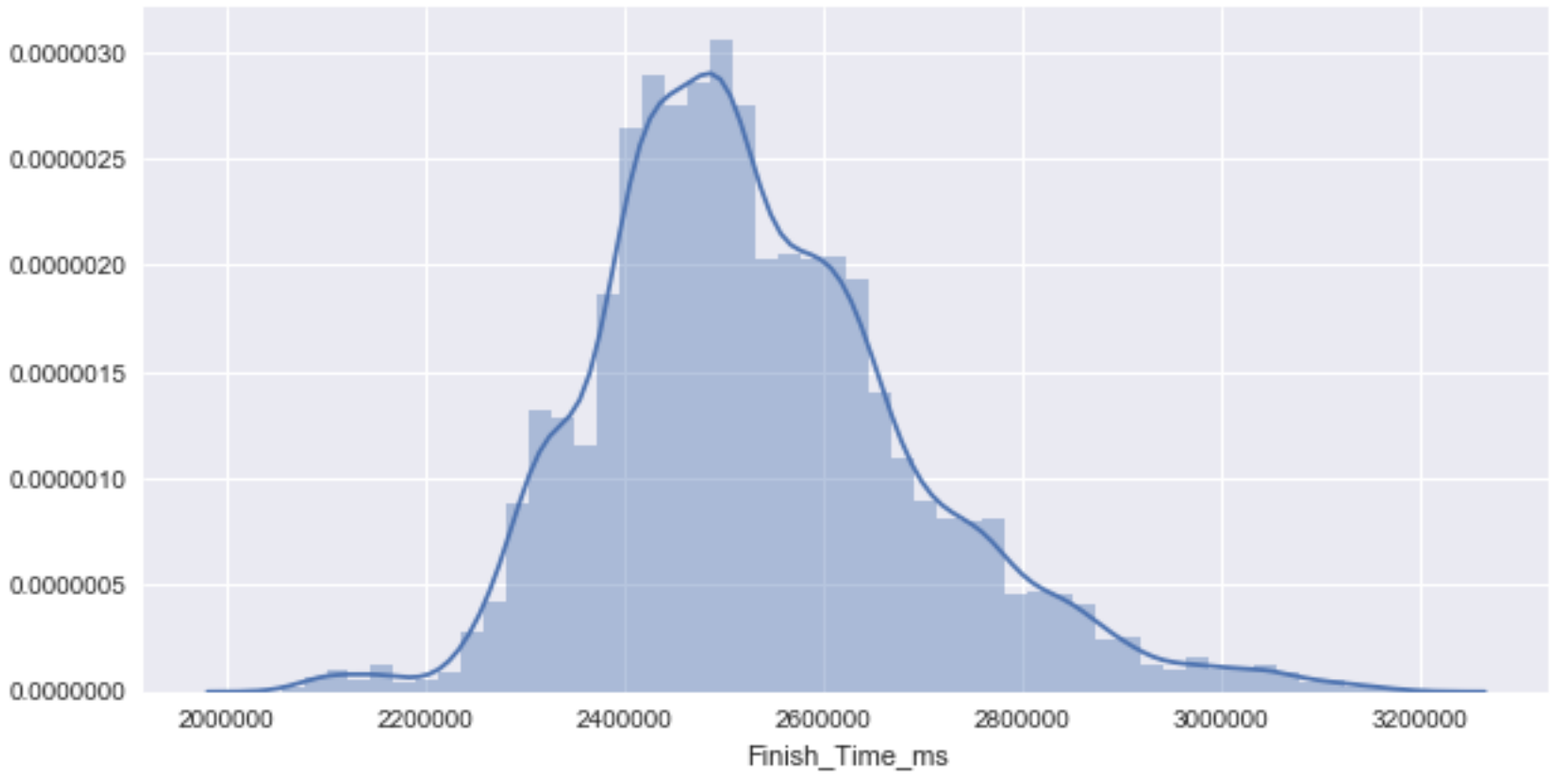
- Plotting the target distribution



- Races that were cut short due to weather, accidents, technical malfunctions, etc.

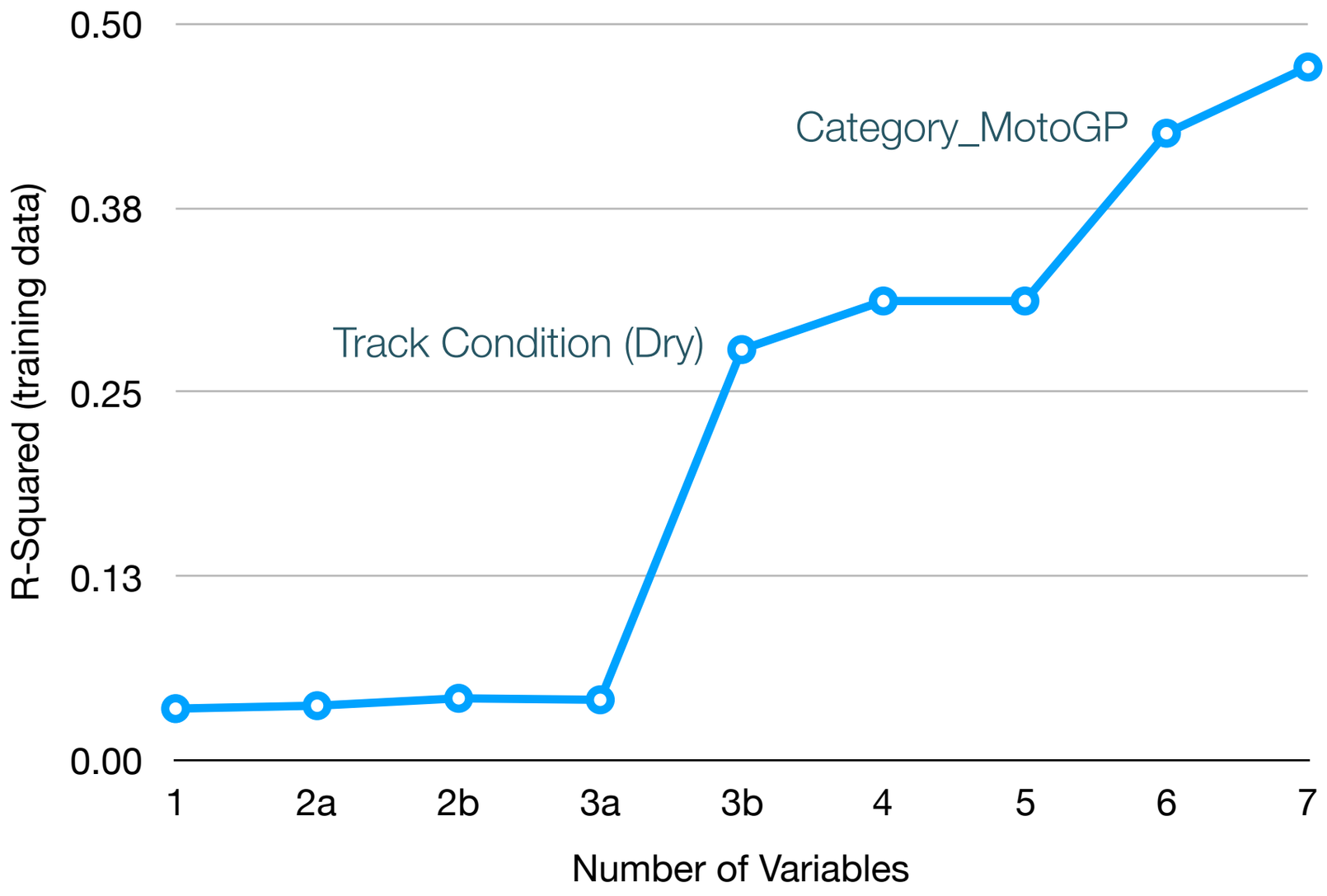
Step 1 - Cleaning

- Plotting the target distribution



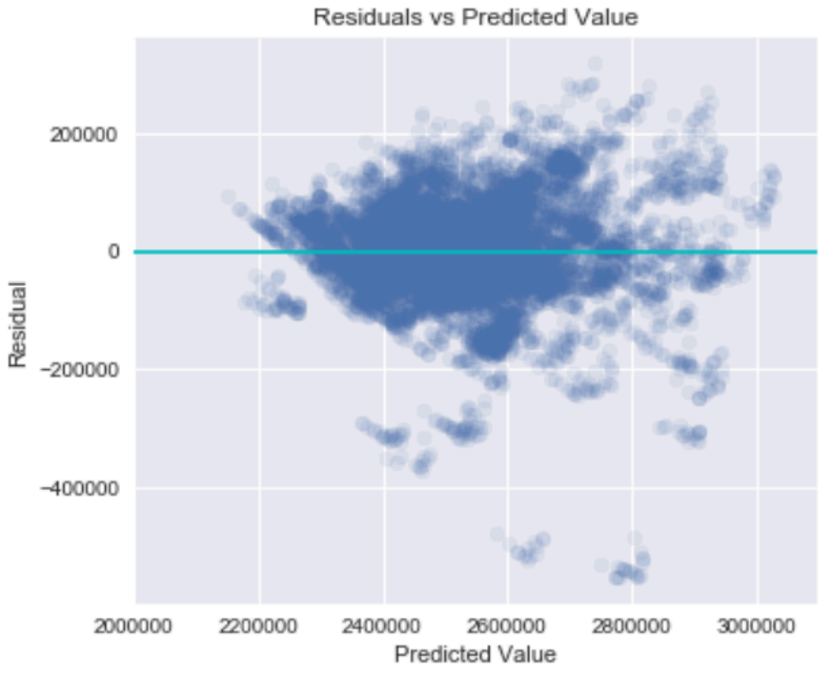
- Much better.

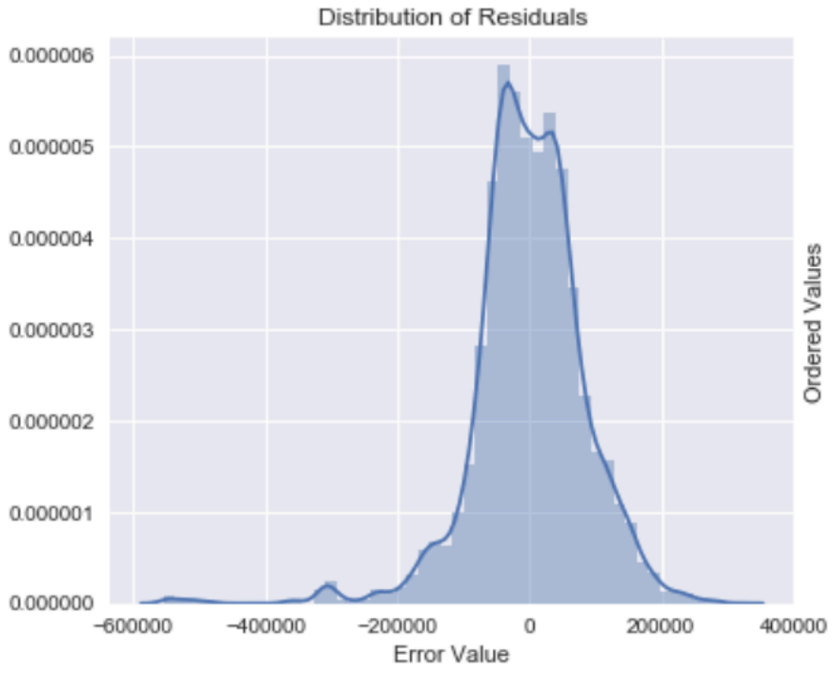
Step 2 - OLS Linear Regression

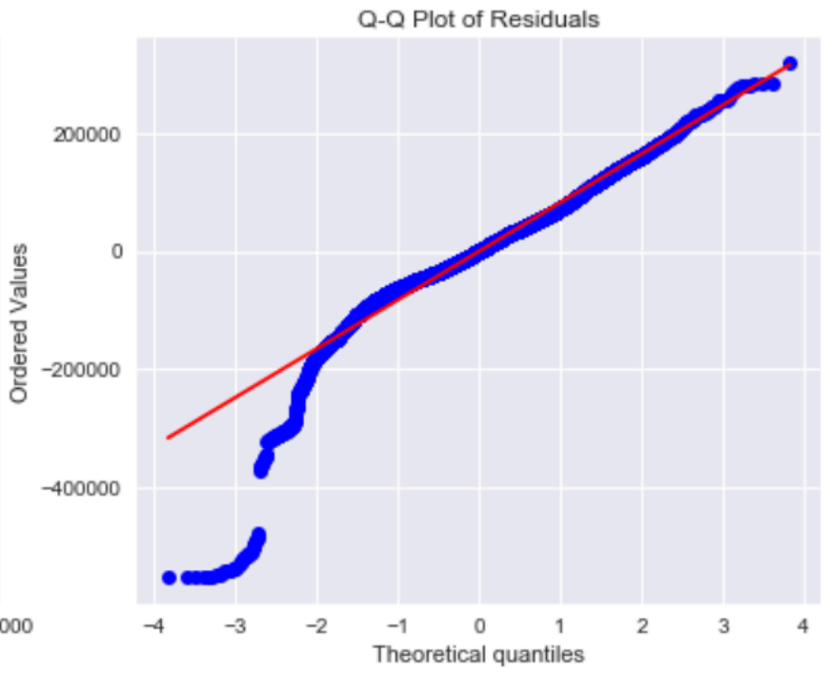


Step 2 - OLS Linear Regression

- Optimized with 16 features
- Test data $R^2 = 0.53$
- Mean RMSE = 110,698 milliseconds (~2 minutes)





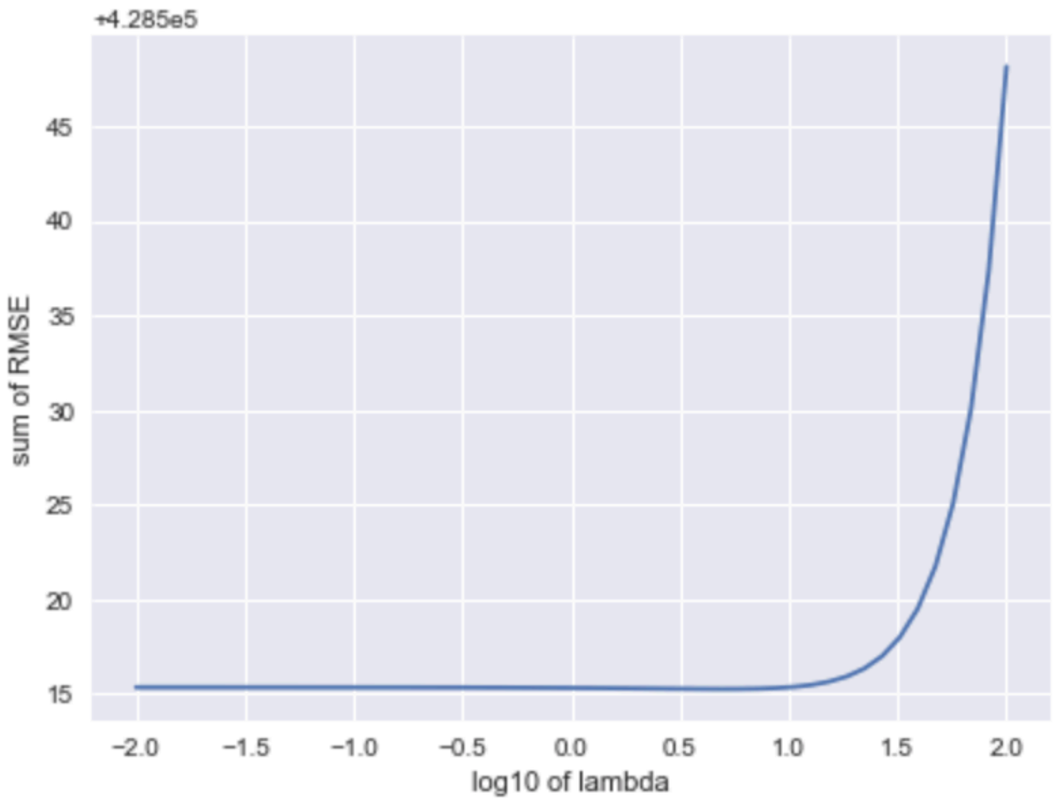


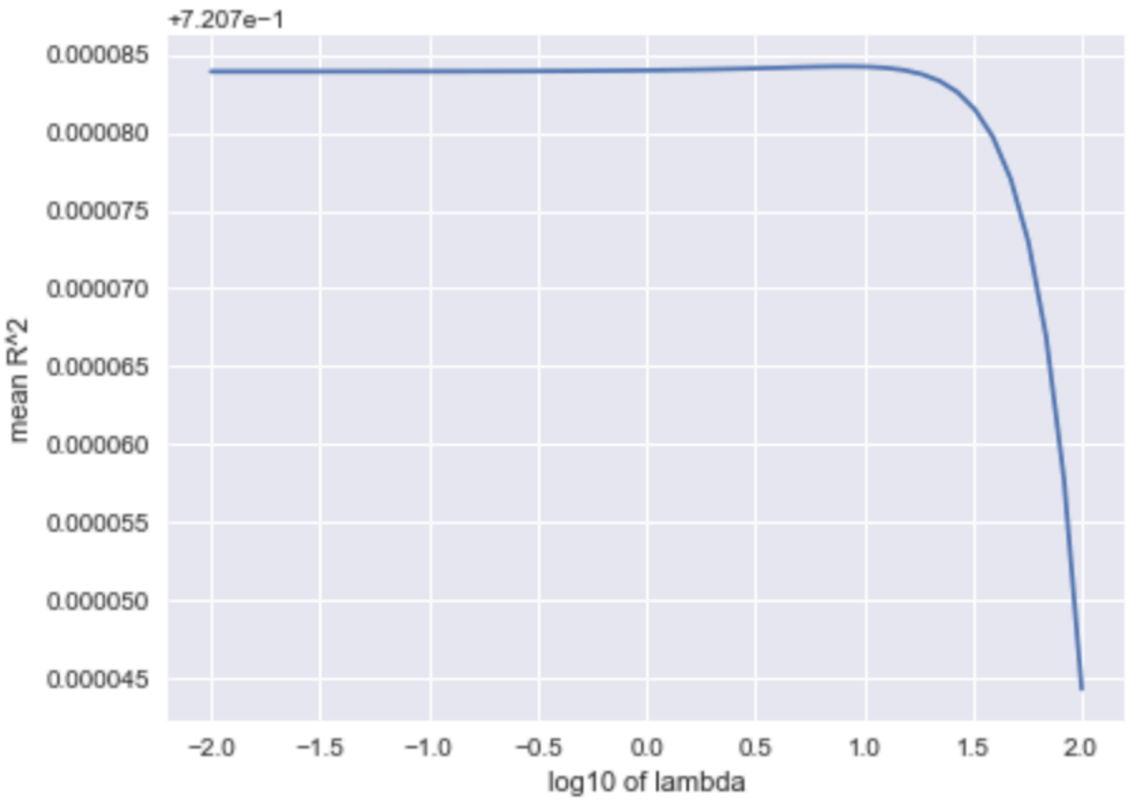
Step 3 - Regularization

- Perhaps lasso regularization with varying λ can help minimize the test error.

Step 3 - Regularization

- Perhaps lasso regularization with varying λ can help minimize the test error.
- Did not drop any features.



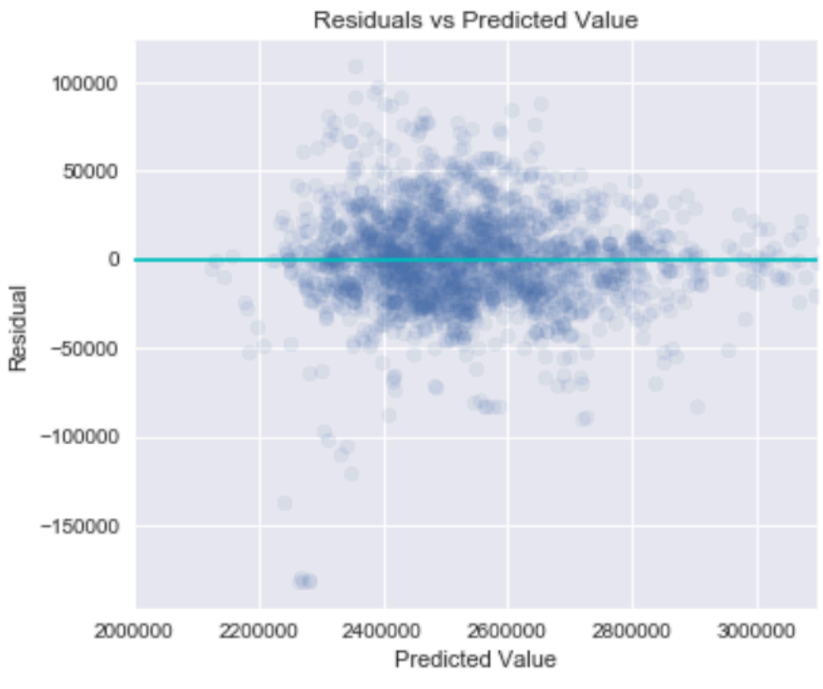


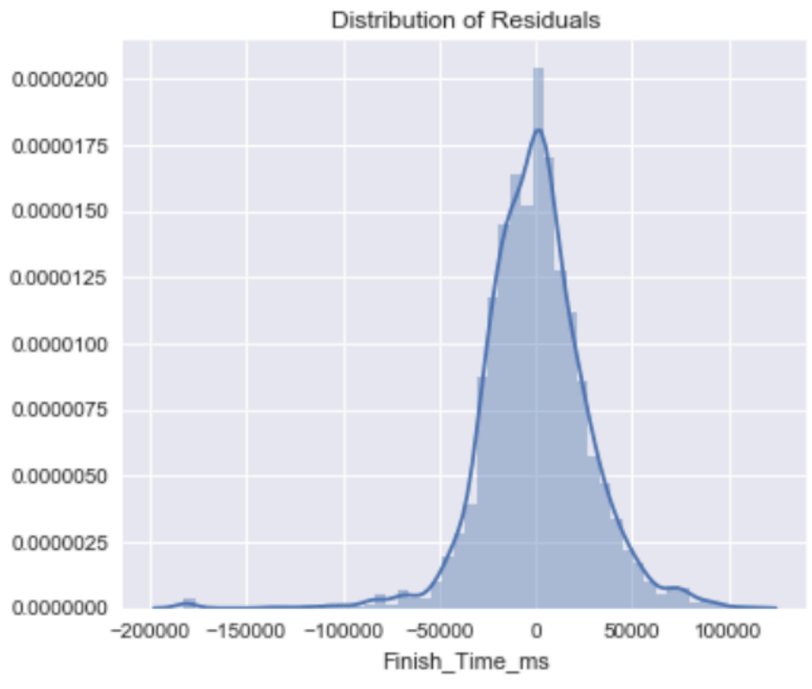
Step 4 - Polynomial Regression

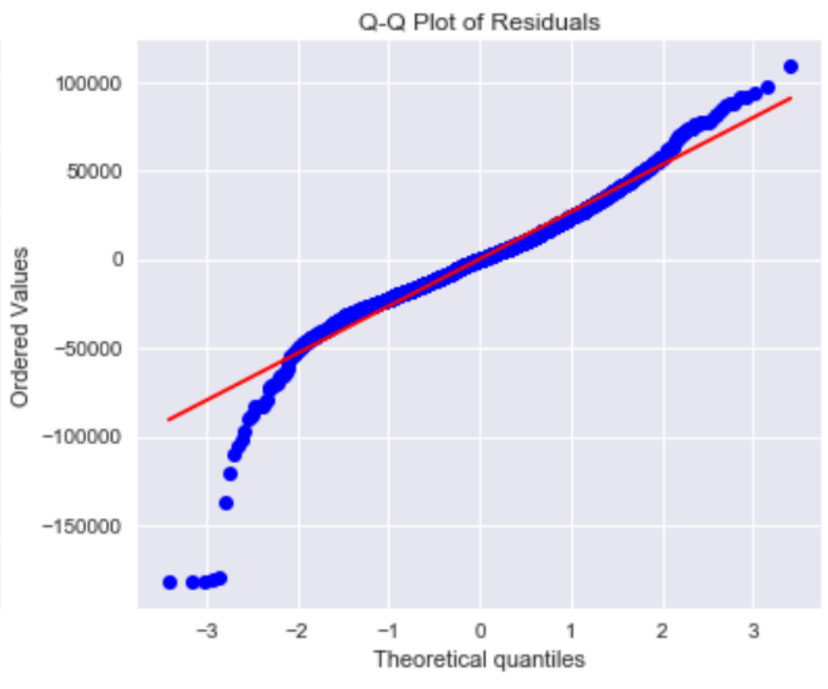
- Can’t improve by only using linear functions of the predictor variables.
- Perhaps the model needs to be more complex.
- After many approaches, settled on a polynomial regression of **degree 3**.

Step 4 - Polynomial Regression

- Test data $R^2 = 0.95$
- Mean RMSE = 36,908 milliseconds
- Typical race = 44 minutes (~ 1.4% error)
- Errors are much smaller in magnitude with this model







Conclusion

- **It is possible** to predict race finish times.
- Simple linear regression is not the best approach.
- Mean RMSE of 37 seconds is problematic.
  - MotoGP is extremely close racing.
  - Final race of last season (2017 Valencia), the top 12 finishers out of 17 finished within 37 seconds of each other.
- Great R^2 value! Error is too high for this application.
- Coefficients are not interpretable.