Predicting Software Development Task Times

Rahul Sane

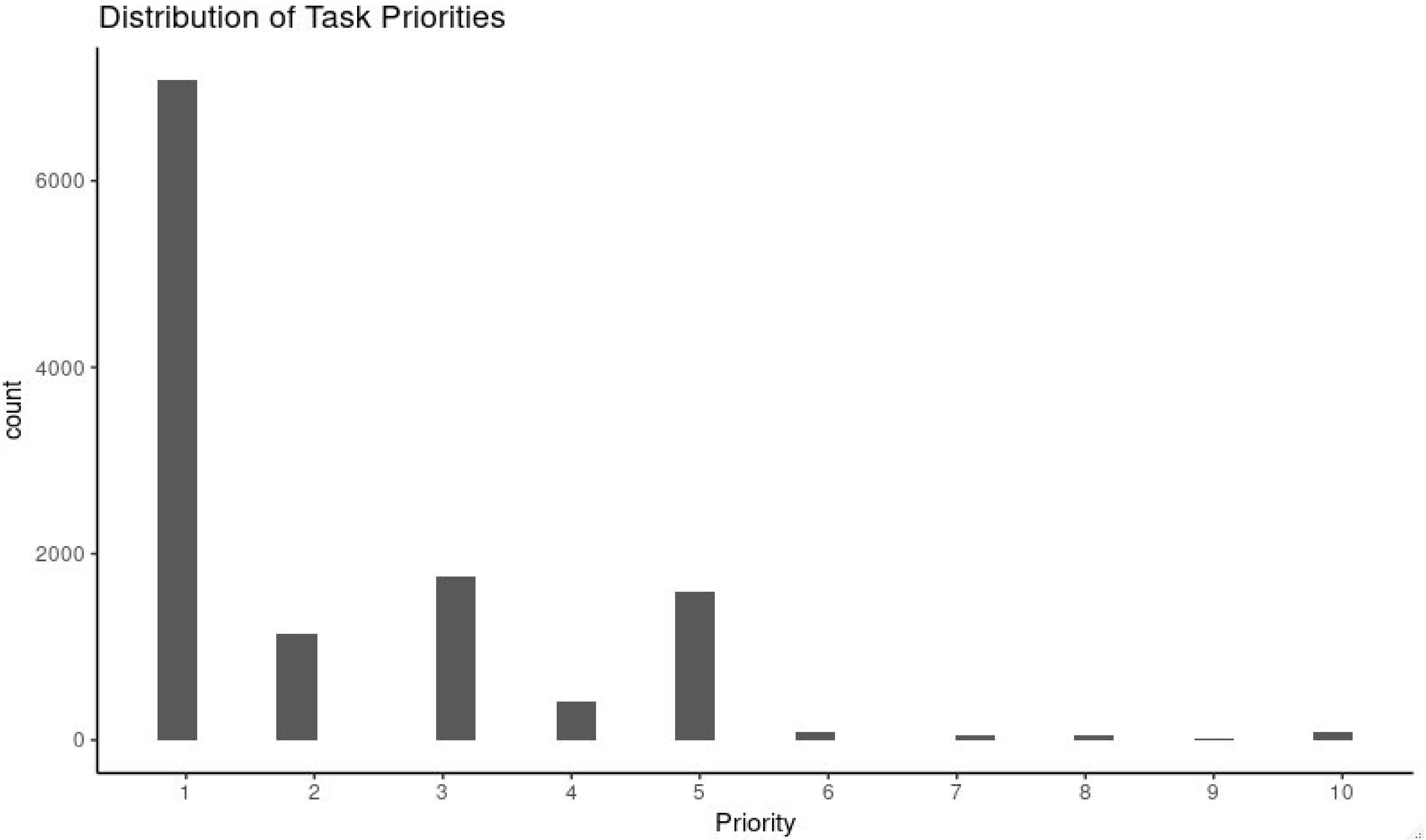
Springboard Intro to Data Science

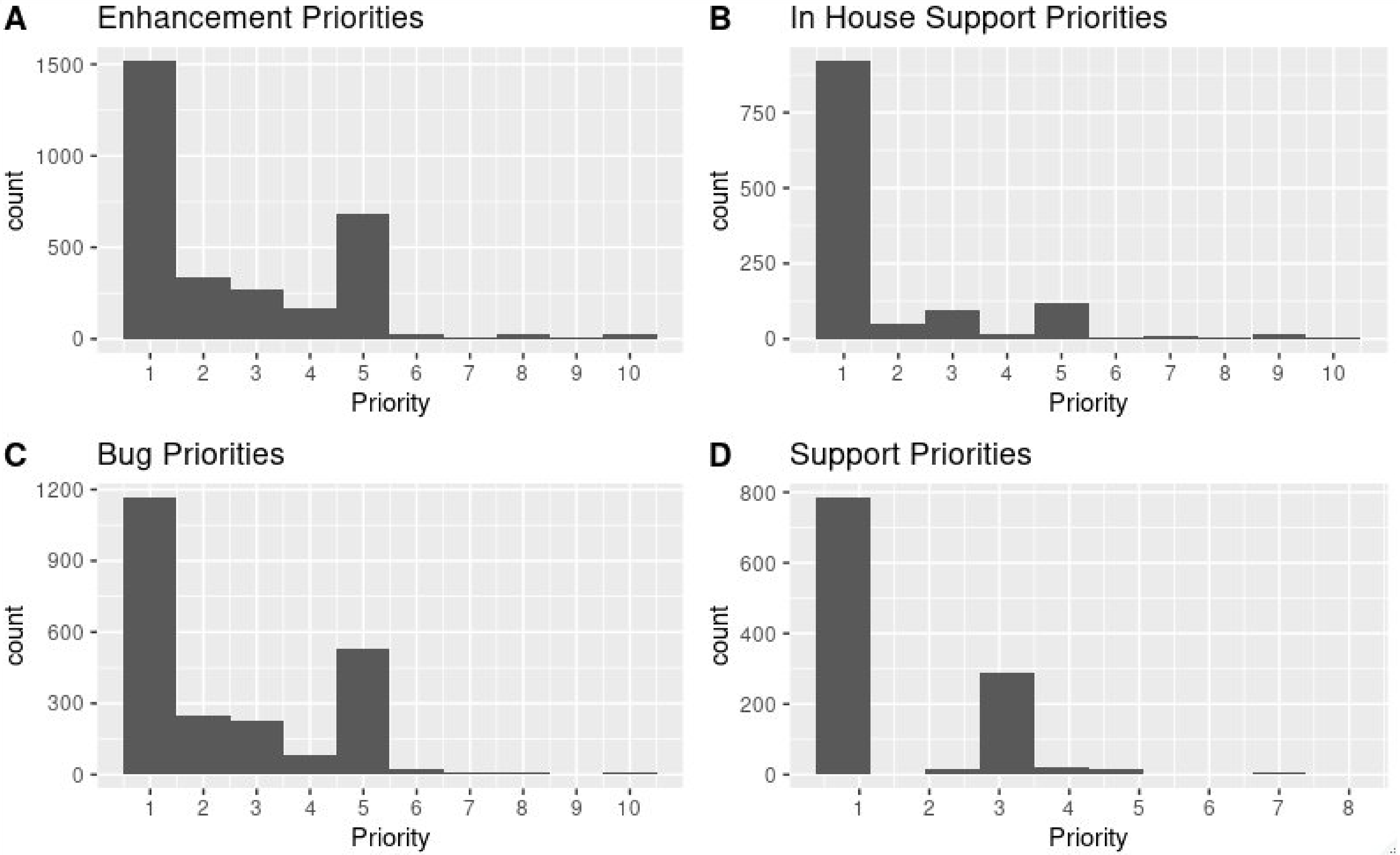
The Data

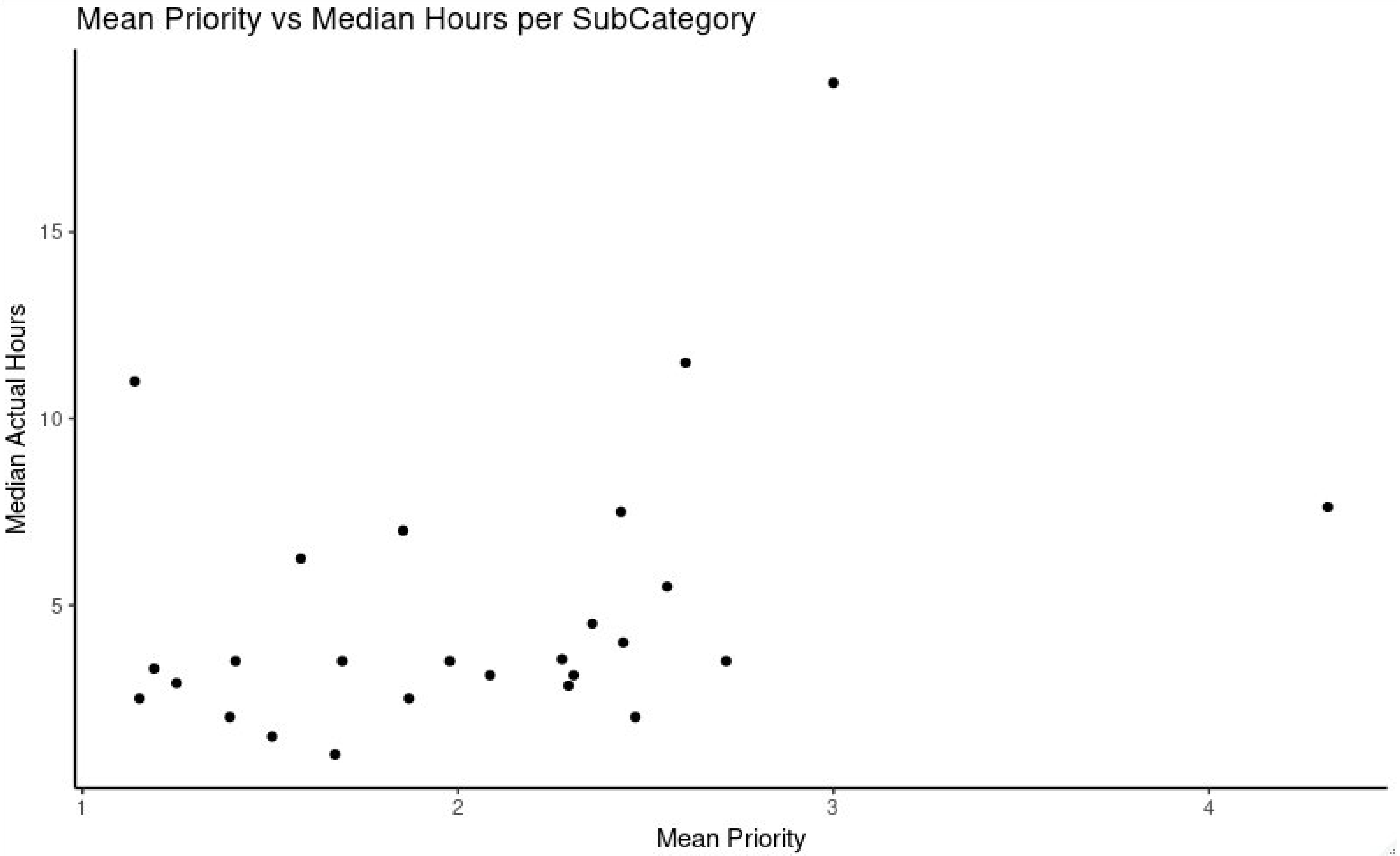
- Historical performance of software developers by task
- Spans 10 years and over 10,000 tasks
- Derek Jones: https://github.com/Derek-Jones/SiP_dataset

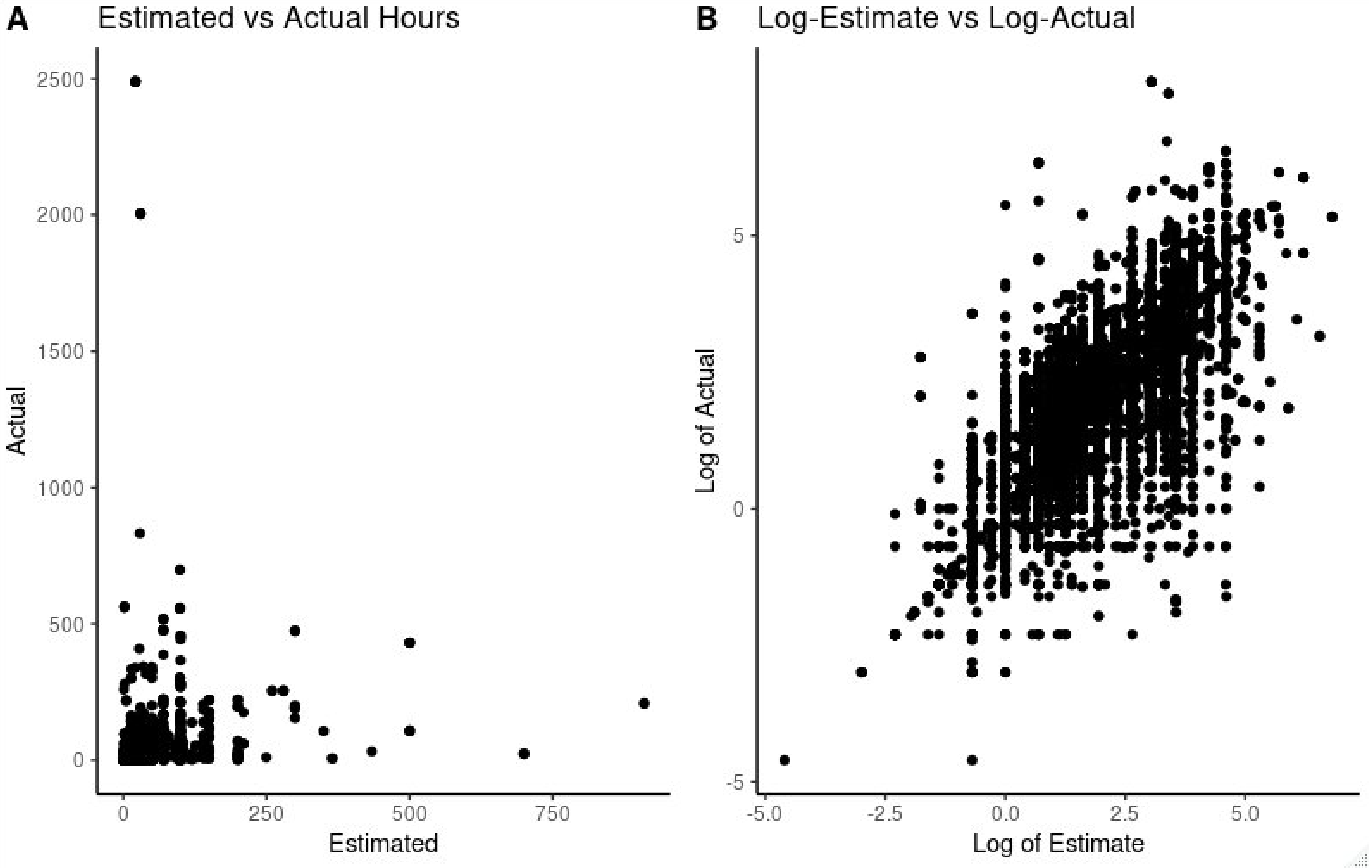
The Data

- Key columns:
  - TaskNumber
  - Summary
  - Category
  - SubCategory
  - HoursEstimate
  - HoursActual
  - DeveloperHoursActual

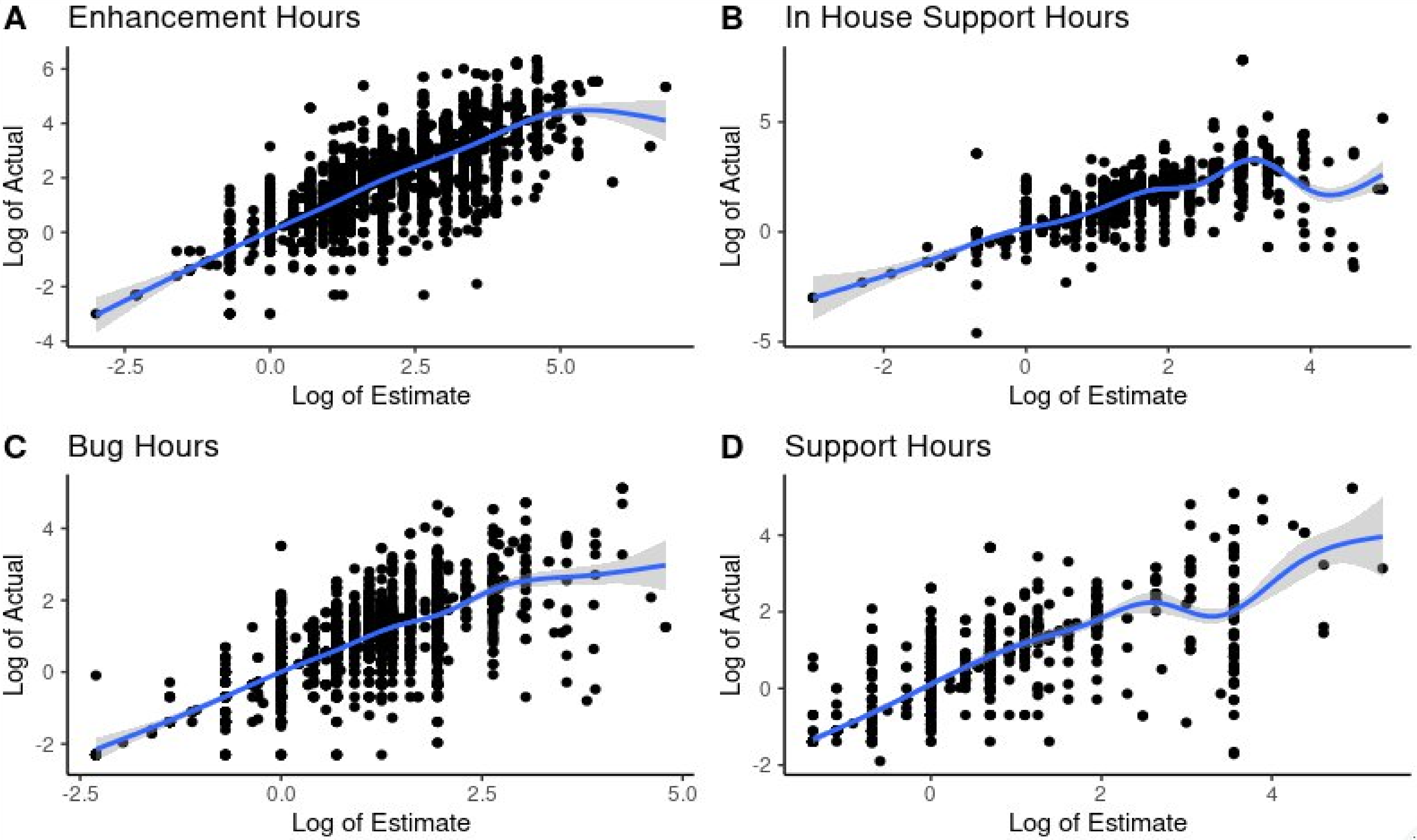

Distribution of Task Priorities
count
6000
4000
2000
0
1
2
3
4
5
6
7
8
9
10
Priority


A
Enhancement Priorities
count
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
Priority
B
In House Support Priorities
count
750
500
250
0
1
2
3
4
5
6
7
8
9
10
Priority
C
Bug Priorities
count
1200
900
600
300
0
1
2
3
4
5
6
7
8
9
10
Priority
D
Support Priorities
count
800
600
400
200
0
1
2
3
4
5
6
7
8
Priority


Mean Priority vs Median Hours per SubCategory
Median Actual Hours
15
10
5
1
2
3
4
Mean Priority



A
Estimated vs Actual Hours
Actual
2500
2000
1500
1000
500
0
0
250
500
750
Estimated
B
Log-Estimate vs Log-Actual
Log of Actual
5
0
-5
-5.0
-2.5
0.0
2.5
5.0
Log of Estimate


A
Enhancement Hours
Log of Actual
Log of Estimate
6
4
2
0
-2
-4
-2.5
0.0
2.5
5.0
B
In House Support Hours
Log of Actual
Log of Estimate
5
0
-5
-2
0
2
4
C
Bug Hours
Log of Actual
Log of Estimate
4
2
0
-2
-2.5
0.0
2.5
5.0
D
Support Hours
Log of Actual
Log of Estimate
4
2
0
-2
0
2
4

Modeling

- Response variable: HoursActual
- Base model using difference between HoursEstimate and HoursActual
- Four machine learning models:
  1. Linear model, HoursEstimate as only predictor
  2. LInear model, all predictors (inc. SubCategory and terms in Summary)
  3. Random forest model, HoursEstimate as only predictor
  4. Random forest model, all predictors

Model Performance

- Root-mean-square error (RMSE)
- Normalized root-mean-square error (NRMSE); expressed as percentage

Model Evaluation

Model	RMSE
Base	27.76
Model 1	19.78
Model 2	19.85
Model 3	20.42
Model 4	19.47

Model Evaluation

Model	RMSE	NRMSE
Model 1	19.78	5.39%
Model 2	19.85	5.41%
Model 3	20.42	5.57%
Model 4	19.47	5.31%

Model Evaluation

- All machine learning models outperform base model
- Simple linear model (Model 2) performed very well
- Model 4 (most complex) is best-performing
  - Most computationally expensive (2.5+ hour runtime)

Use Case

- Company that relies on internal softwares
- Improve developer workflows
- Improve project management
- Tighter possible timelines

Further Investigations

- Improve prediction accuracy
- Topic modeling based on summary terms
- Clustering using columns not used here
- Other ML models, inc. SVM and polynomial regression
- Need to avoid over-fitting with more complex models