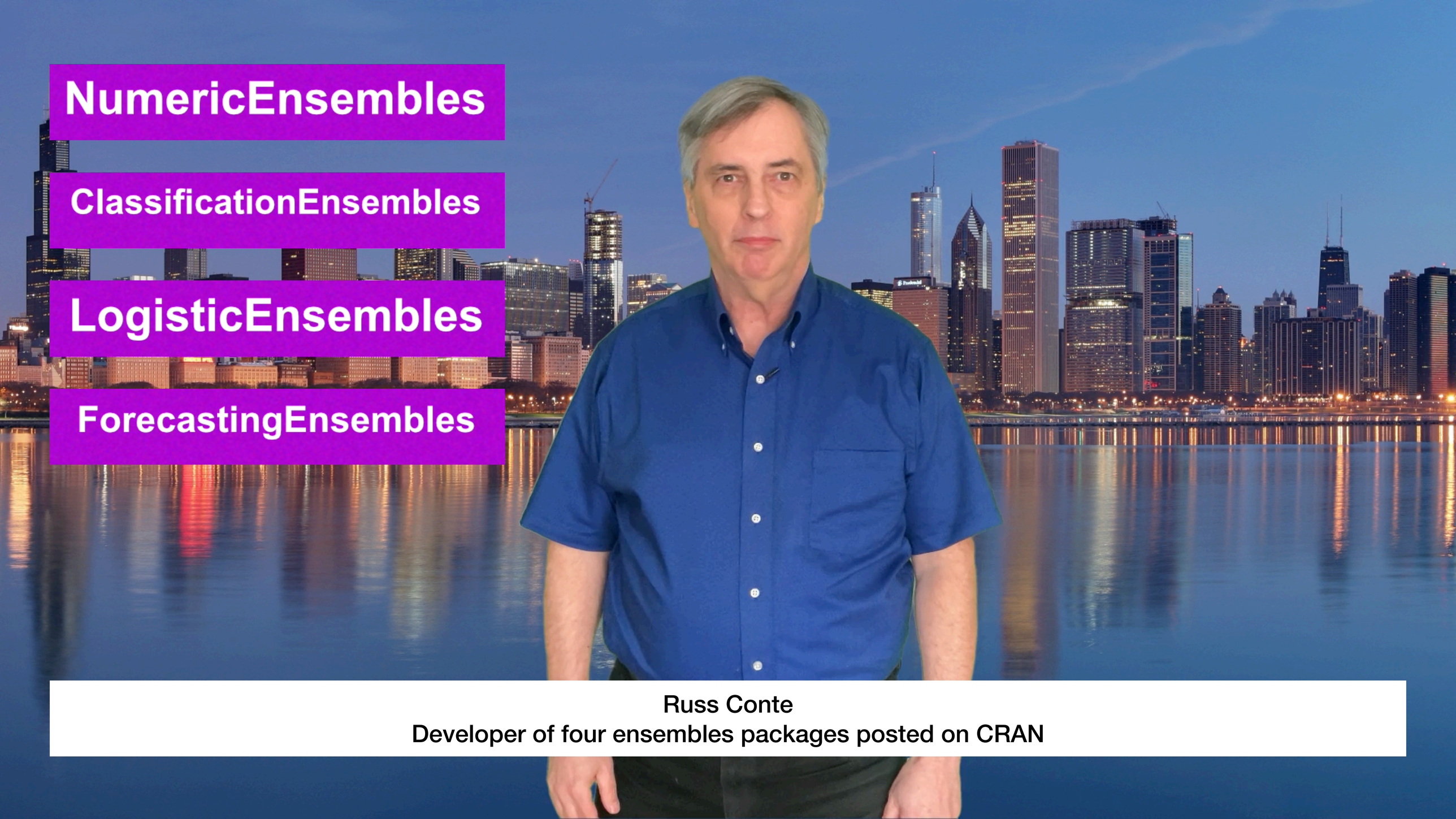


Russ Conte
Developer of four ensembles packages posted on CRAN

All source code (including examples) is available at:

GitHub.com/InfiniteCuriosity

ensembles1.com

NumericEnsembles.com
ClassificationEnsembles.com
LogisticEnsembles.com
ForecastingEnsembles.com

russconte@mac.com

NumericEnsembles (on CRAN)
ClassificationEnsembles (on CRAN)
LogisticEnsembles (on CRAN)
ForecastingEnsembles (on CRAN)

Outline:

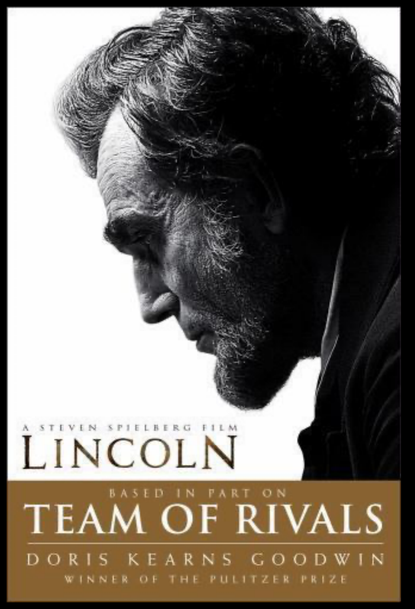
1. Intro to NumericEnsembles with the Insurance data set
2. Intro to LogisticEnsembles with the Pima Indians diabetes data set
3. Intro to ClassificationEnsembles with the Cleveland heart risk data set
4. Summary/wrap–up
5. Q&A

(What is one change/improvement you would like to see in any of the packages?)

Note: This is ALL designed to solve REAL business problems.

1. Introduction to NumericEnsembles with the Insurance data set:

Meet the 32 numeric models!



18 individual models
14 ensembles

Here is your first team of rivals!

NumericEnsembles list of 32 models

Model Name	Individual	Ensembles
Bagging	✔	✔
BayesGLM	✔	✔
BayesRNN	✔	✔
Cubist	✔	✔
Earth	✔	✔
Elastic	✔	✔
Generalized Additive Models	✔	✘
Gradient Boosted	✔	✔
Lasso	✔	✔
Linear	✔	✔
Neuralnet	✔	✔
Principal Components Regression	✔	✘
Partial Least Squares	✔	✘
Ridge	✔	✔
RPart	✔	✔
Support Vector Machines	✔	✔
Tree	✔	✔
XGBoost	✔	✘
Total	18	14

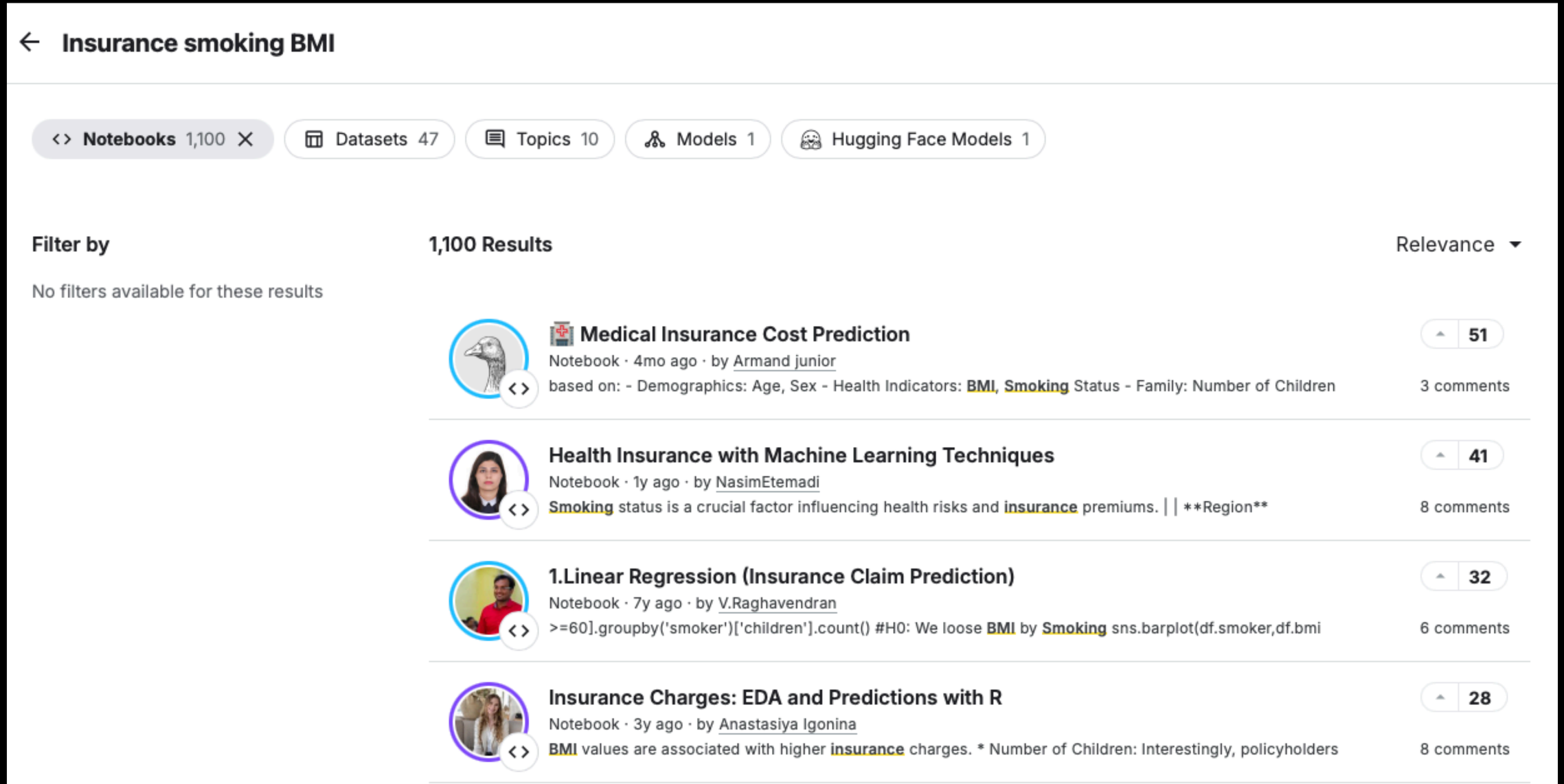
Let’s look at the Insurance data set (first ten rows shown)

	age	sex	bmi	children	smoker	region	charges
1	19	female	27.900	0	yes	southwest	16884.924
2	18	male	33.770	1	no	southeast	1725.552
3	28	male	33.000	3	no	southeast	4449.462
4	33	male	22.705	0	no	northwest	21984.471
5	32	male	28.880	0	no	northwest	3866.855
6	31	female	25.740	0	no	southeast	3756.622
7	46	female	33.440	1	no	southeast	8240.590
8	37	female	27.740	3	no	northwest	7281.506
9	37	male	29.830	2	no	northeast	6406.411
10	60	female	25.840	0	no	northwest	28923.137

Let’s look at six sample results for this data set

Name	Lowest RMSE	Source
Insurance EDA + Smoking & BMI Impact + Prediction	$4,329.57	https://www.kaggle.com/code/alexkhr/insurance-eda-smoking-bmi-impact-prediction
MedCost: EDA|K-Cluster🎛️|📚Gradient Boost [Full]	$4,352.03	https://www.kaggle.com/code/tumpanjawat/medcost-eda-k-cluster-gradient-boost-full
Health Insurance with Machine Learning Techniques	$4,621.23	https://www.kaggle.com/code/nasimetemadi/health-insurance-with-machine-learning-techniques
Medical Cost Prediction	$4,951.04	https://www.kaggle.com/code/tahaahmedt/medical-cost-prediction
Health Care Cost Prediction w/ Linear Regression	$5,642.45	https://www.kaggle.com/code/ruslankl/health-care-cost-prediction-w-linear-regression
Insurance Premium	$6,005.59	https://www.kaggle.com/code/venky12347/insurance-premium

Over 1,000 results for this data set on kaggle.com



NumericEnsembles example:

Accuracy by model

Overfitting by model

Separators/risk in the original data

Many more results are all delivered automatically

```
library(NumericEnsembles)
Numeric(data = Insurance,
        colnum = 7,
        numresamples = 10,
        remove_VIF_above = 5.00,
        remove_data_correlations_greater_than = 0.99,
        remove_ensemble_correlations_greater_than = 1.00,
        scale_all_predictors_in_data = "N",
        data_reduction_method = 1,
        ensemble_reduction_method = 0,
        how_to_handle_strings = 2,
        predict_on_new_data = "N",
        save_all_trained_models = "N",
        stratified_random_sampling = "N",
        set_seed = "N",
        save_all_plots = "N",
        use_parallel = "Y",
        train_amount = 0.60,
        test_amount = 0.20,
        validation_amount = 0.20)
```

```
library(NumericEnsembles)
Numeric(data = Insurance,
        colnum = 7,
        numresamples = 10,
        remove_VIF_above = 5.00,
        remove_data_correlations_greater_than = 0.99,
        remove_ensemble_correlations_greater_than = 1.00,
        scale_all_predictors_in_data = "N",
        data_reduction_method = 1,
        ensemble_reduction_method = 0,
        how_to_handle_strings = 2,
        predict_on_new_data = "N",
        save_all_trained_models = "N",
        stratified_random_sampling = "N",
        set_seed = "N",
        save_all_plots = "N",
        use_parallel = "Y",
        train_amount = 0.60,
        test_amount = 0.20,
        validation_amount = 0.20)
```

Numeric(data = Insurance,
colnum = 7,
numresamples = 10,

```
library(NumericEnsembles)
Numeric(data = Insurance,
        colnum = 7,
        numresamples = 10,
        remove_VIF_above = 5.00,
        remove_data_correlations_greater_than = 0.99,
        remove_ensemble_correlations_greater_than = 1.00,
        scale_all_predictors_in_data = "N",
        data_reduction_method = 1,
        ensemble_reduction_method = 0,
        how_to_handle_strings = 2,
        predict_on_new_data = "N",
        save_all_trained_models = "N",
        stratified_random_sampling = "N",
        set_seed = "N",
        save_all_plots = "N",
        use_parallel = "Y",
        train_amount = 0.60,
        test_amount = 0.20,
        validation_amount = 0.20)
```

```
library(NumericEnsembles)
Numeric(data = Insurance,
        colnum = 7,
        numresamples = 10,
        remove_VIF_above = 5.00,
        remove_data_correlations_greater_than = 0.99,
        remove_ensemble_correlations_greater_than = 1.00,
        scale_all_predictors_in_data = "N",
        data_reduction_method = 1,
        ensemble_reduction_method = 0,
        how_to_handle_strings = 2,
        predict_on_new_data = "N",
        save_all_trained_models = "N",
        stratified_random_sampling = "N",
        set_seed = "N",
        save_all_plots = "N",
        use_parallel = "Y",
        train_amount = 0.60,
        test_amount = 0.20,
        validation_amount = 0.20)
```


```
remove_VIF_above = 5.00,
remove_data_correlations_greater_than = 0.99,
remove_ensemble_correlations_greater_than = 1.00,
scale_all_predictors_in_data = "N",
data_reduction_method = 1,
ensemble_reduction_method = 0,
how_to_handle_strings = 2,
predict_on_new_data = "N",
save_all_trained_models = "N",
stratified_random_sampling = "N",
set_seed = "N",
save_all_plots = "N",
use_parallel = "Y",
```

RStudio

Go to file/function | Addins | Project: (None)

_NumericEnsembles template.R

Source on Save | Run | Source

```r
#install.packages(NumericEnsembles)

start_time <- Sys.time()

library(NumericEnsembles)
Numeric(data = Insurance,
  colnum = 7,
  numresamples = 25,
  remove_VIF_above = 5.00,
  remove_data_correlations_greater_than = 0.99,
  remove_ensemble_correlations_greater_than = 1.00,
  scale_all_predictors_in_data = "N",
  data_reduction_method = 0,
  ensemble_reduction_method = 0,
  how_to_handle_strings = 1,
  predict_on_new_data = "N",
  save_all_trained_models = "N",
  stratified_random_sampling = "N",
  set_seed = "N",
  save_all_plots = "N",
  use_parallel = "Y",
  train_amount = 0.60,
  test_amount = 0.20,
  validation_amount = 0.20)

  end_time <- Sys.time()
  duration <- end_time - start_time
  duration
  warnings()

tempdir()
```

13:29 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.5.2 · ~/Library/Mobile Documents/com~apple~CloudDocs/Documents/Machine Learning templates in R/

```
R version 4.5.2 (2025-10-31) -- "[Not] Part in a Rumble"
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Environment | History | Connections | Tutorial

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export

RMSE, means, fitting, model summaries of the train, test and validation sets

	Model	Mean_holdout_RMSE	RMSE_Lower_95_Conf_Int	RMSE_Upper_95_Conf_Int	RMSE_Std_Dev	Overfitting_mean	Overfitting_lower_95_CI	Overfitting_upper_95_CI	Overfitting_sd
1	Actual data	0	0	0	0	0	0	0	0
2	Ensemble Earth	82.1009	75.0482	89.1535	4.637	0.8971	0.8201	0.9742	0.055
3	Ensemble Neuralnet	88.9341	81.2944	96.5737	0.1576	1.0417	0.9522	1.1312	0.0871
4	Ensemble BayesGLM	88.9419	81.3016	96.5822	0.1481	1.0422	0.9527	1.1318	0.0878
5	BayesRNN	100.8353	92.1733	109.4972	2.275	0.9705	0.8872	1.0539	0.0215
6	Ensemble Cubist	189.0294	172.7913	205.2674	8.4373	3.6297	3.3179	3.9415	0.6235
7	Ensemble BayesRNN	193.7051	177.0654	210.3449	2.1124	1.4159	1.2942	1.5375	0.0342
8	Ensemble Lasso	365.7667	334.3465	397.187	3.6157	0.9958	0.9103	1.0814	0.1965
9	Ensemble Elastic	476.3291	435.4112	517.2469	130.3612	0.9437	0.8626	1.0247	0.2088
10	Ensemble Gradient Boosted	829.5285	758.27	900.787	87.5109	1.0111	0.9242	1.0979	0.0801

RMSE, means, fitting, model summaries of the train, test and validation sets

	Model	Mean_holdout_RMSE	RMSE_Lower_95_Conf_Int	RMSE_Upper_95_Conf_Int	RMSE_Std_Dev	Overfitting_mean	Overfitting_lower_95_CI	Overfitting_upper_95_CI	Overfitting_sd
1	Actual data	0	0	0	0	0	0	0	0
2	Ensemble Earth	82.1009	75.0482	89.1535	4.637	0.8971	0.8201	0.9742	0.055
3	Ensemble Neuralnet	88.9341	81.2944	96.5737	0.1576	1.0417	0.9522	1.1312	0.0871
4	Ensemble BayesGLM	88.9419	81.3016	96.5822	0.1481	1.0422	0.9527	1.1318	0.0878
5	BayesRNN	100.8353	92.1733	109.4972	2.275	0.9705	0.8872	1.0539	0.0215
6	Ensemble Cubist	189.0294	172.7913	205.2674	8.4373	3.6297	3.3179	3.9415	0.6235
7	Ensemble BayesRNN	193.7051	177.0654	210.3449	2.1124	1.4159	1.2942	1.5375	0.0342
8	Ensemble Lasso	365.7667	334.3465	397.187	3.6157	0.9958	0.9103	1.0814	0.1965
9	Ensemble Elastic	476.3291	435.4112	517.2469	130.3612	0.9437	0.8626	1.0247	0.2088
10	Ensemble Gradient Boosted	829.5285	758.27	900.787	87.5109	1.0111	0.9242	1.0979	0.0801

14	SVM	1402.7674	1282.2663	1523.2685	329.0353	1.3681	1.2506	1.4857	0.0155	
15	Ensemble Linear	2030.9955	1856.5281	2205.4629	28.0805	1.0175	0.9301	1.1049	0.0207	
16	Ensemble Rpart	2030.9955	1856.5281	2205.4629	28.0805	1.0175	0.9301	1.1049	0.0207	
17	Ensemble Trees	2045.7236	1869.991	2221.4562	35.3107	1.0319	0.9433	1.1206	0.0236	
18	Cubist	4520.5278	4132.2036	4908.8521	95.3771	0.9982	0.9125	1.084	0.0137	
19	Gradient Boosted	4682.8635	4280.5942	5085.1328	57.751	1.3629	1.2458	1.48	0.0174	
20	Bagging	4769.2444	4359.5548	5178.934	92.7353	1.0167	0.9294	1.104	0.0121	
21	XGBoost	4772.4787	4362.5112	5182.4461	80.7829	1.4943	1.3659	1.6226	0.0222	
22	Linear	4994.5474	4565.5037	5423.5911	104.3143	1.0196	0.932	1.1071	0.0167	
23	Rpart	4994.5474	4565.5037	5423.5911	104.3143	1.0196	0.932	1.1071	0.0167	
24	Tree	4994.5474	4565.5037	5423.5911	104.3143	1.0196	0.932	1.1071	0.0167	
25	Neuralnet	5993.1396	5478.3144	6507.9647	19.2543	0.9894	0.9044	1.0744	0.0176	
26	Earth	6021.9946	5504.6908	6539.2985	14.3537	1.007	0.9205	1.0935	0.0072	
27	BayesGLM	6031.7835	5513.6387	6549.9283	19.2429	0.9919	0.9067	1.0771	0.0079	
28	Elastic	6032.9585	5514.7128	6551.2042	108.9399	0.9958	0.9103	1.0814	0.197	
29	Lasso	6034.5193	5516.1395	6552.8991	110.4775	0.9959	0.9104	1.0815	0.1971	
30	Ridge	6096.0015	5572.3402	6619.6627	129.1224	0.9983	0.9126	1.0841	0.1975	
31	PLS	8479.261	7750.872	9207.65	65.7637	1.0005	0.9146	1.0865	0.012	
32	PCR	10827.3367	9897.2423	11757.431	141.9699	1.0144	0.9273	1.1016	0.007	
33	GAM	11546.4868	10554.6157	12538.3579	165.085	1.0169	0.9295	1.1042	0.0068	

How does this work?

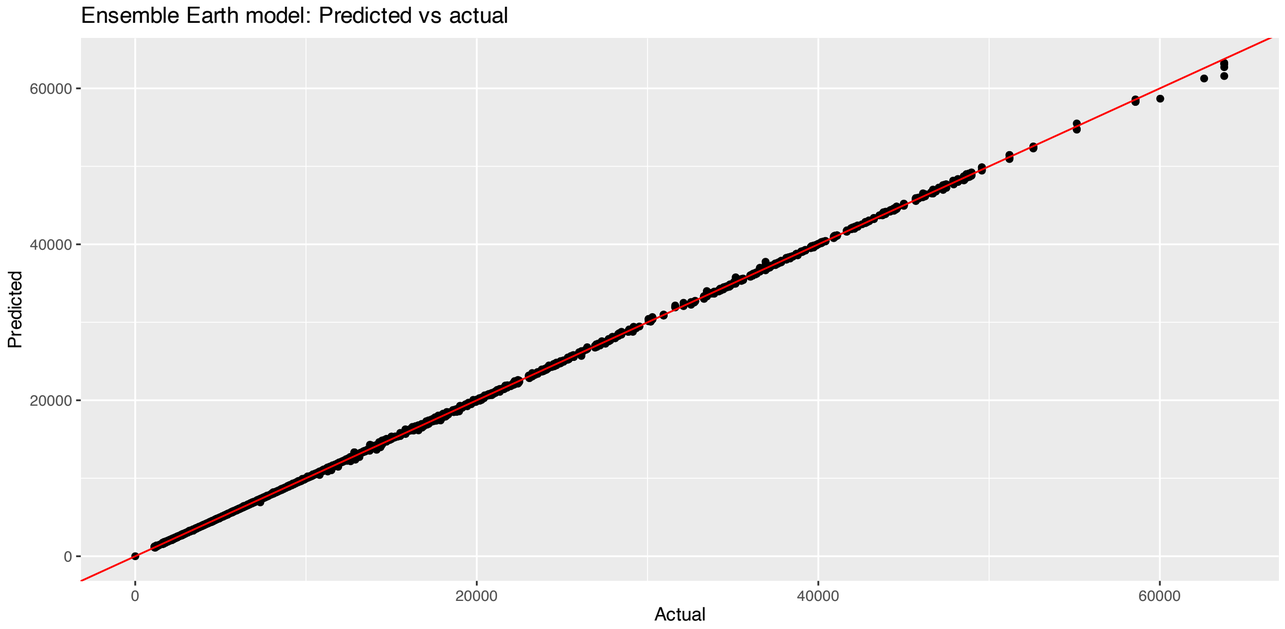
We'll look at Support Vector Machines as a good example

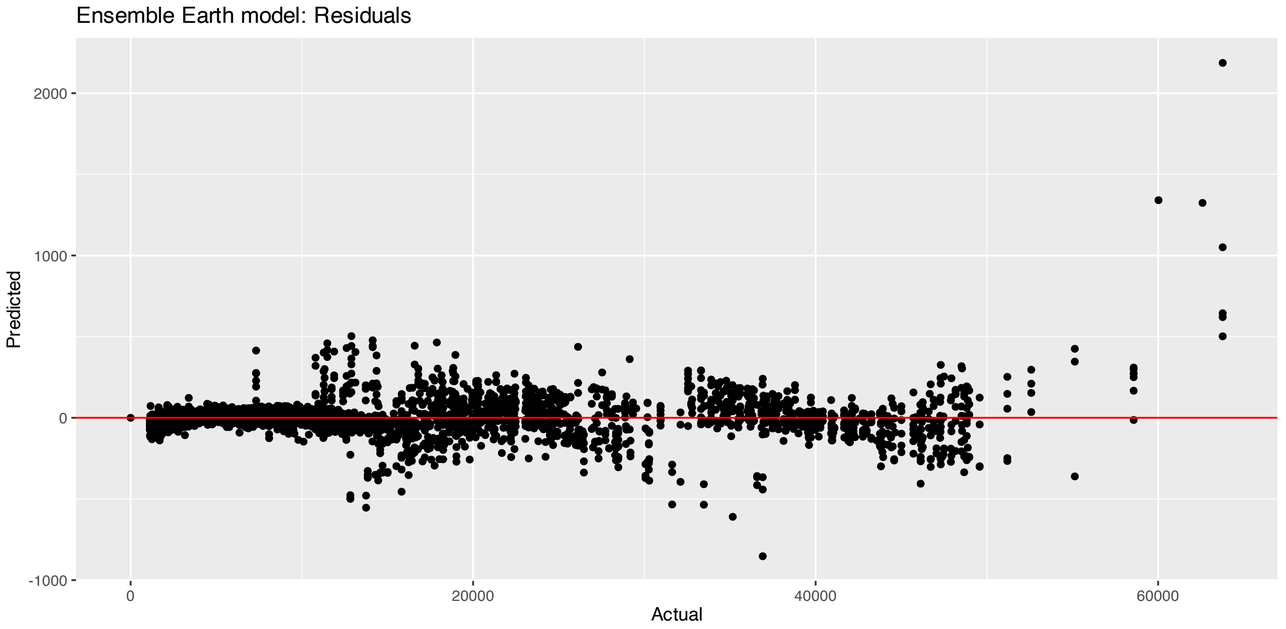
```
svm_train_fit <- e1071::tune.svm(x = train, y = train$y, data = train)
```

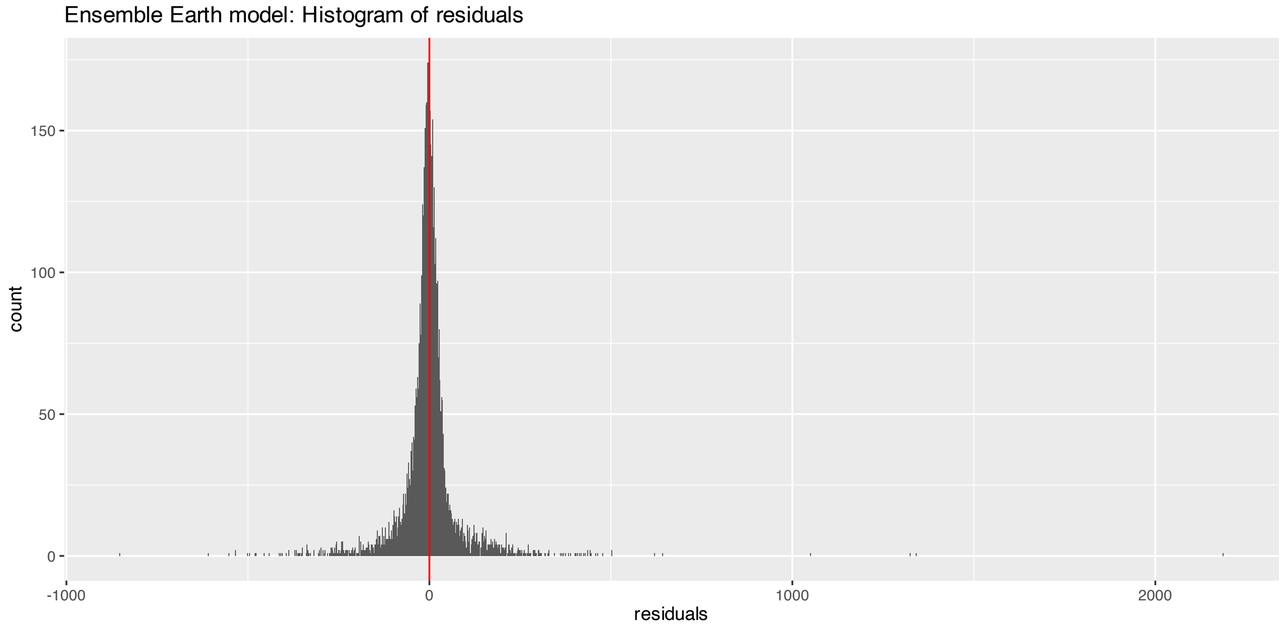
```
svm_test_RMSE[i] <- Metrics::rmse(actual = test$y, predicted = predict(object =
svm_train_fit$best.model, newdata = test))
```

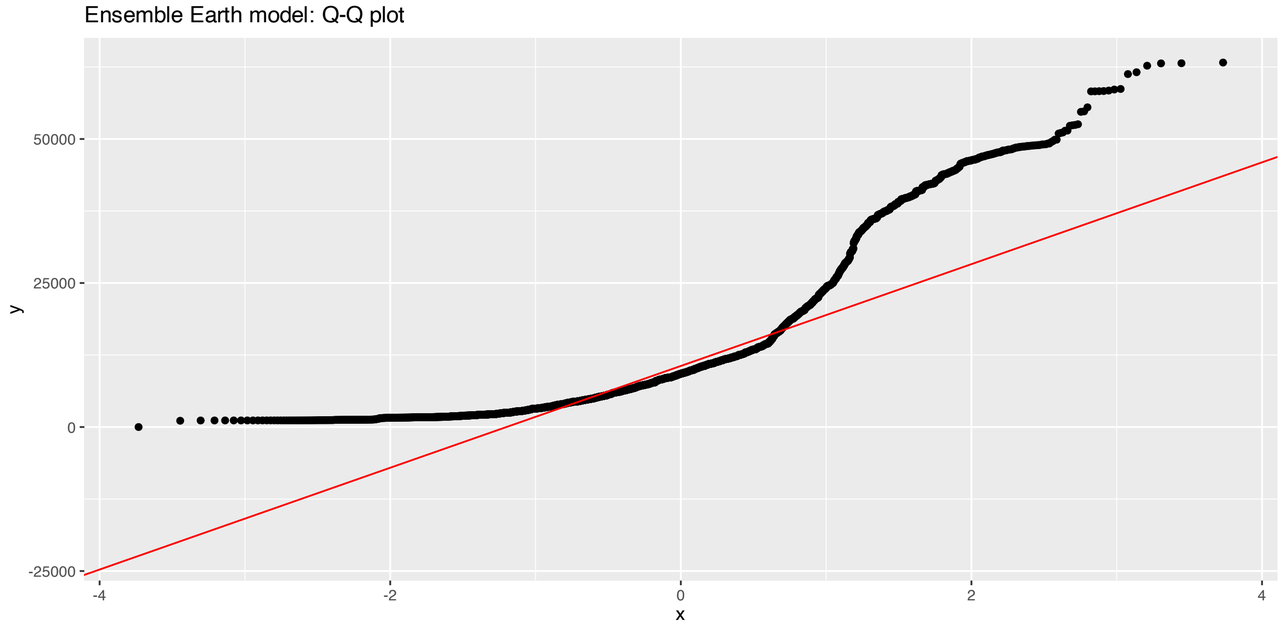
```
svm_test_RMSE_mean <- mean(svm_test_RMSE)
```

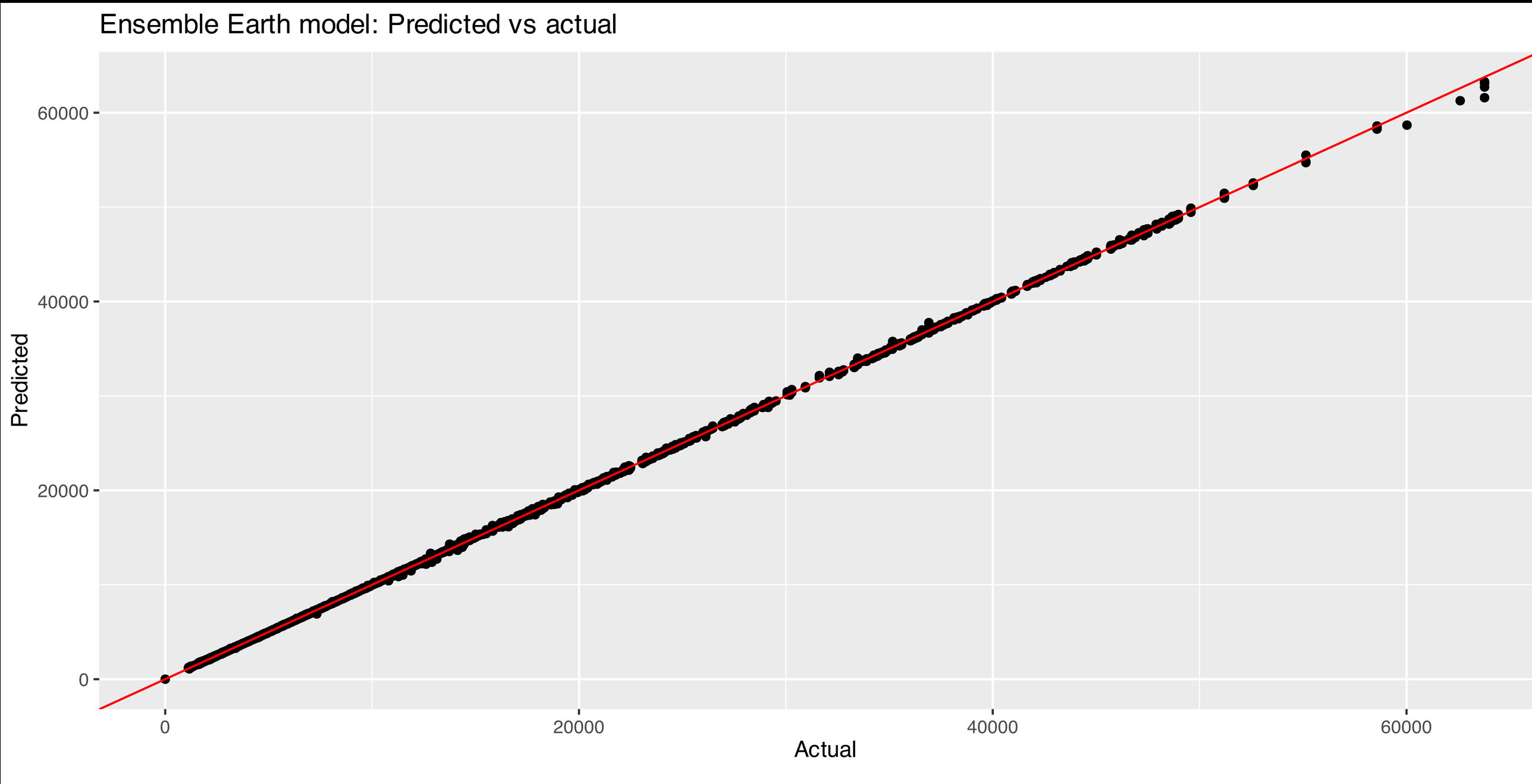
A video of a presentation I gave that has a different perspective

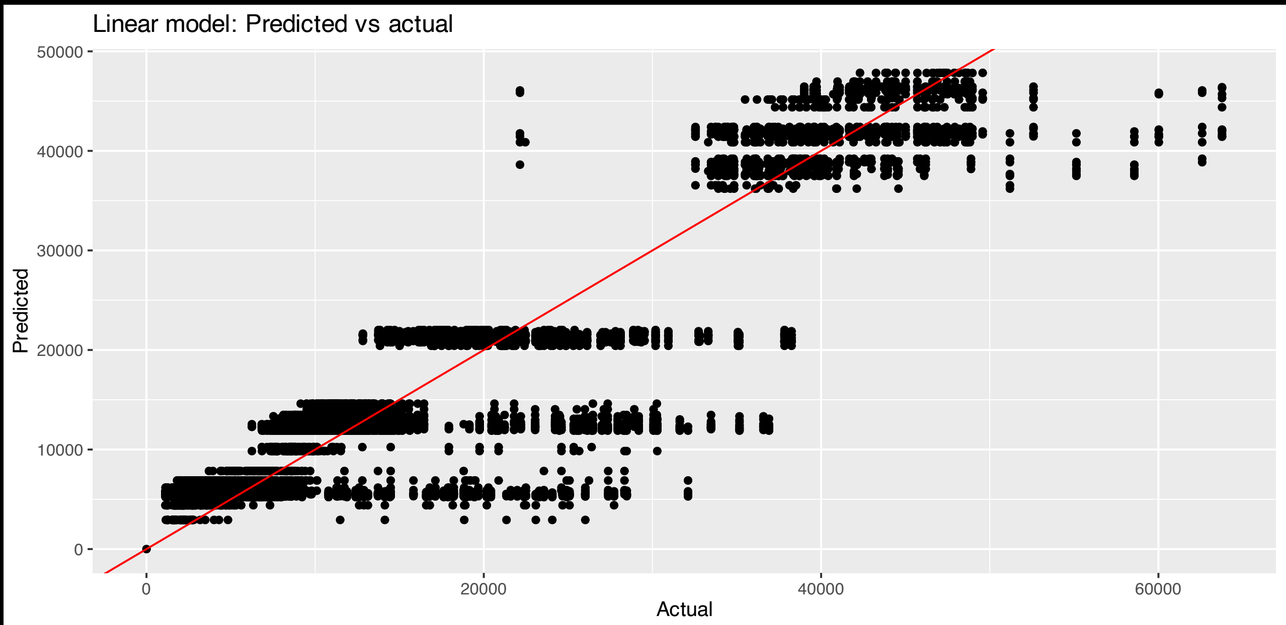

Ensemble Earth model: Predicted vs actual
Predicted
Actual
60000
40000
20000
0
0
20000
40000
60000

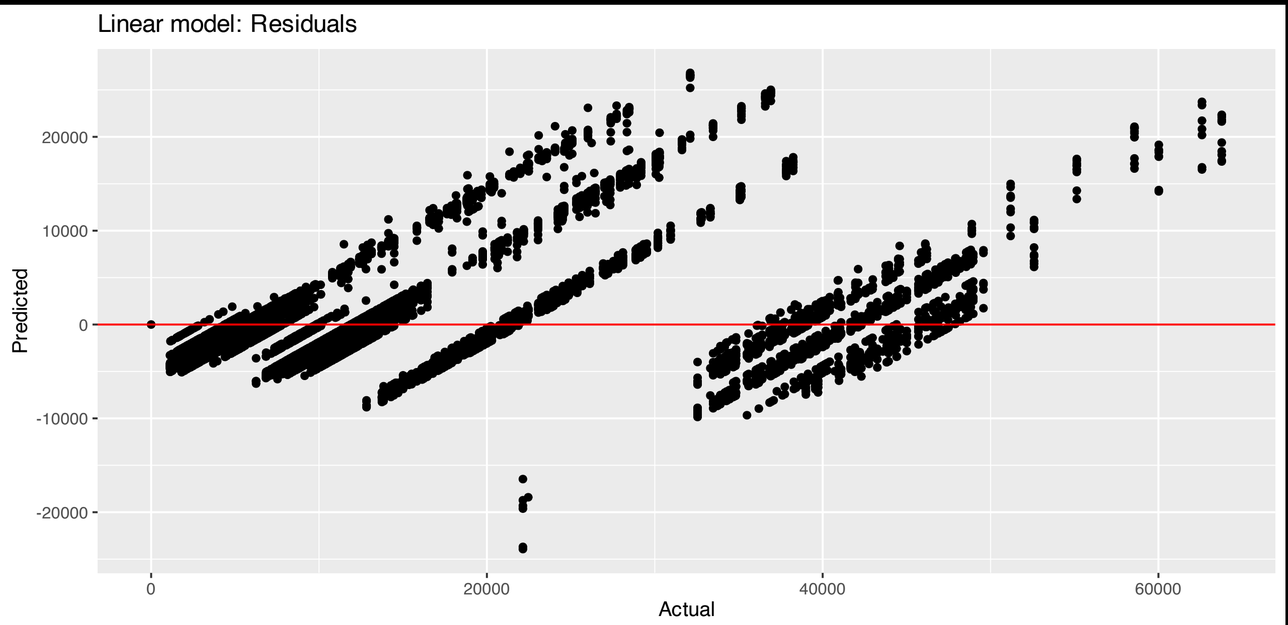

Ensemble Earth model: Residuals
Predicted
Actual
2000
1000
0
-1000
0
20000
40000
60000

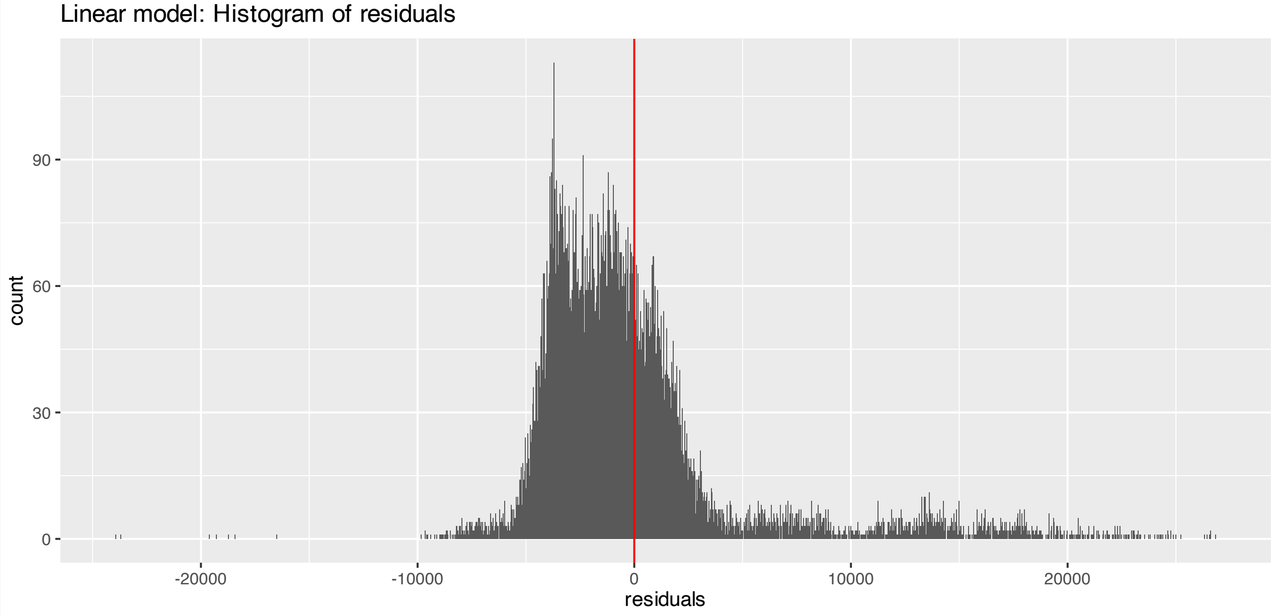

Ensemble Earth model: Histogram of residuals
count
residuals
150
100
50
0
-1000
0
1000
2000

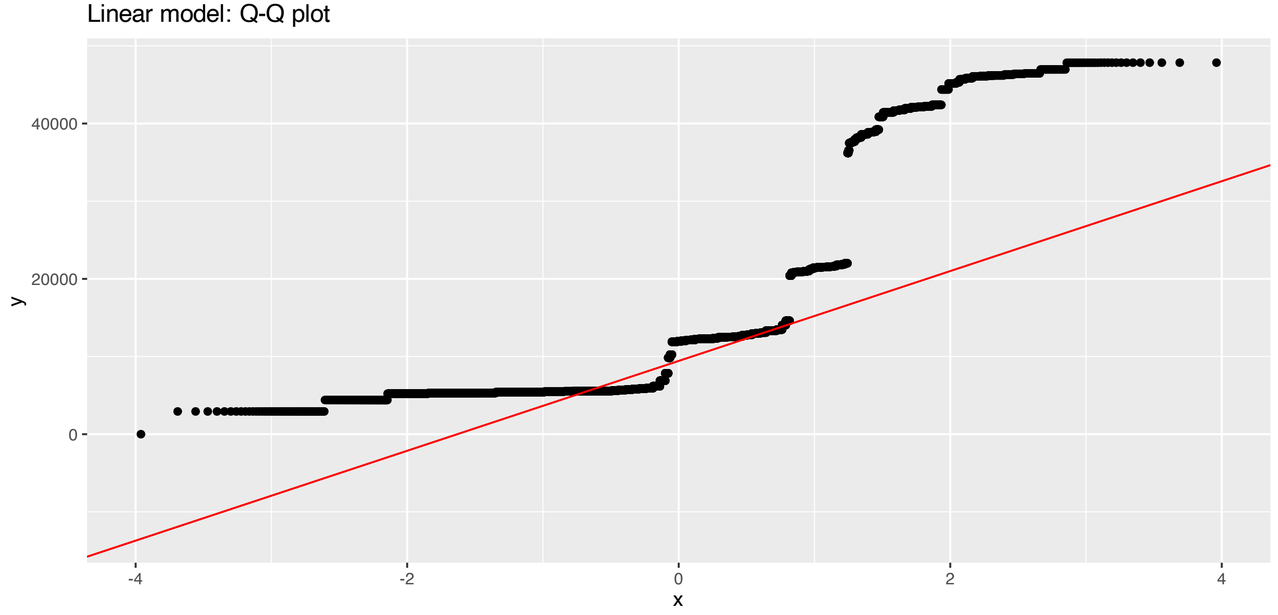

Ensemble Earth model: Q-Q plot
y
x
50000
25000
0
-25000
-4
-2
0
2
4


Ensemble Earth model: Predicted vs actual
60000
40000
20000
0
Predicted
0
20000
40000
60000
Actual

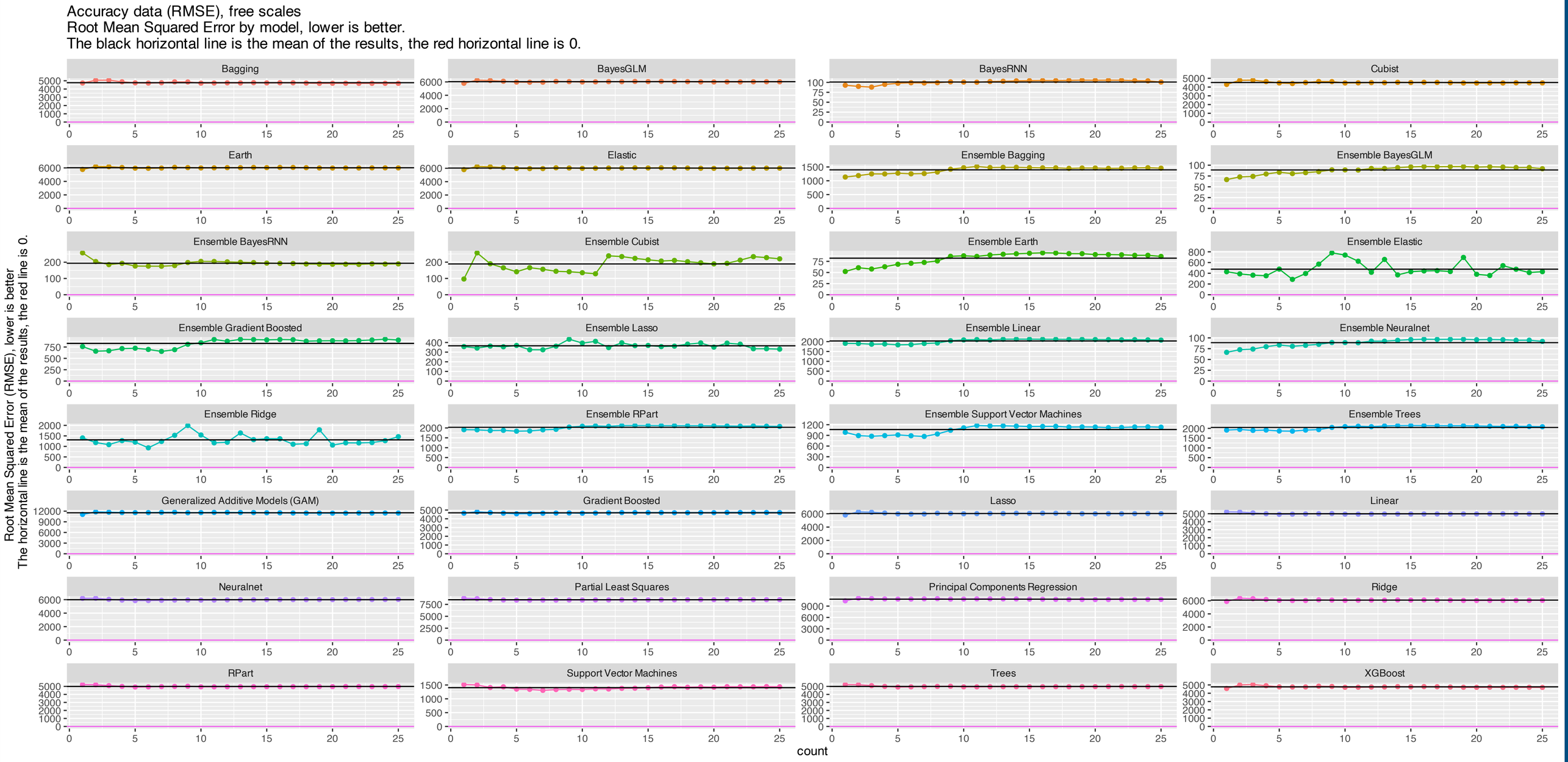

Linear model: Predicted vs actual
Predicted
Actual
50000
40000
30000
20000
10000
0
0
20000
40000
60000


Linear model: Residuals
Predicted
Actual
20000
10000
0
-10000
-20000
0
20000
40000
60000


Linear model: Histogram of residuals
count
residuals
90
60
30
0
-20000
-10000
0
10000
20000

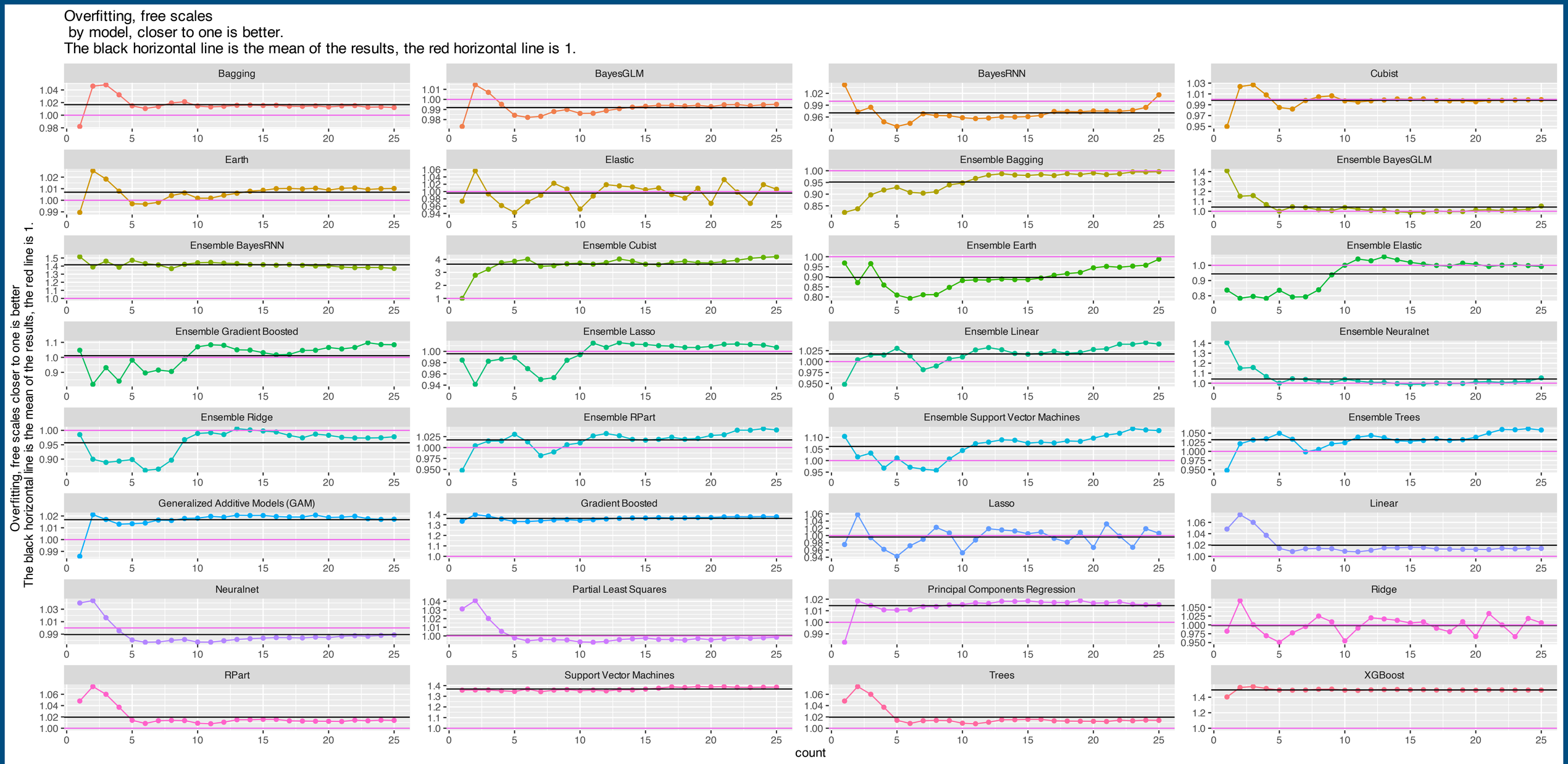

Linear model: Q-Q plot
y
x
40000
20000
0
-4
-2
0
2
4

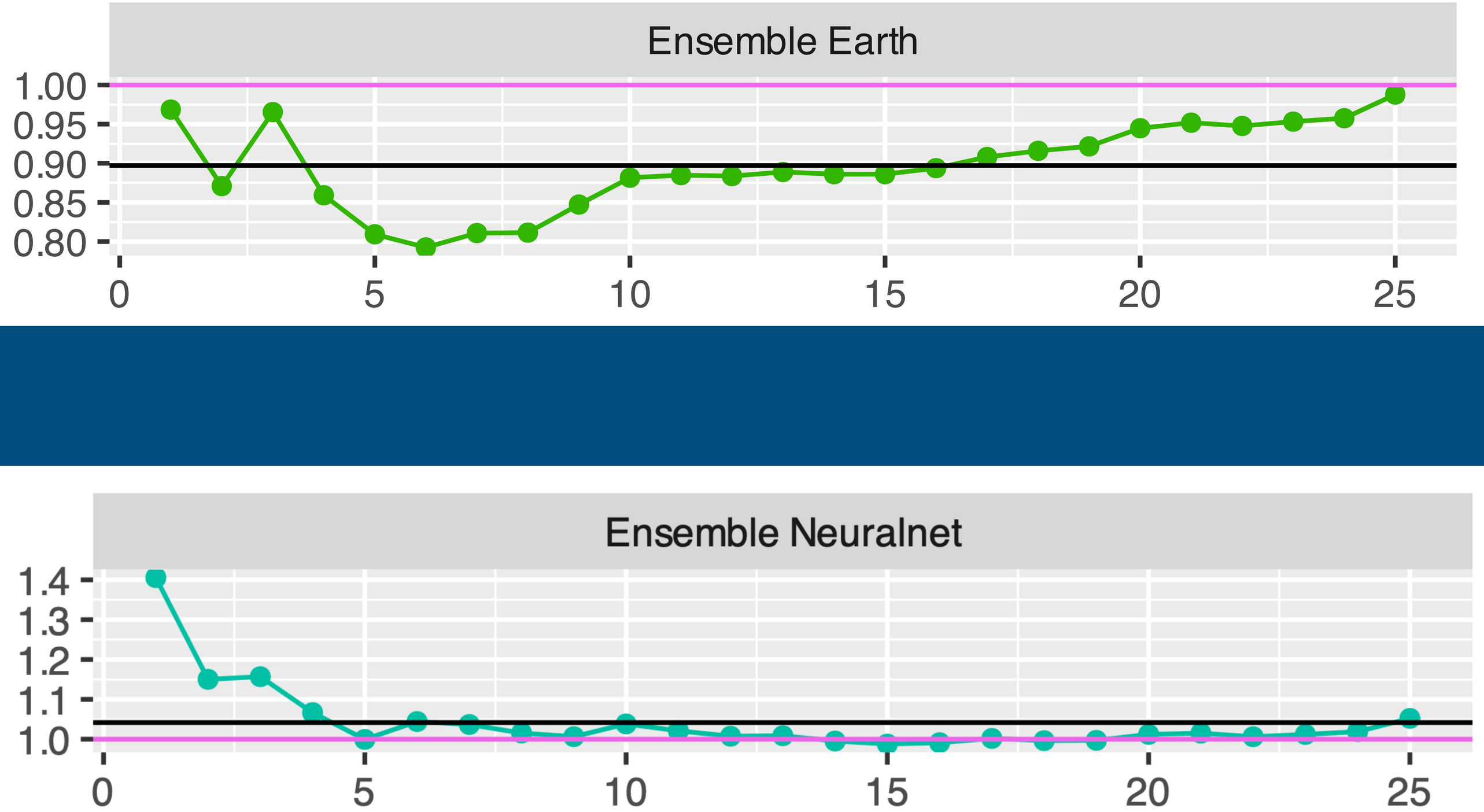
Accuracy by model and resample



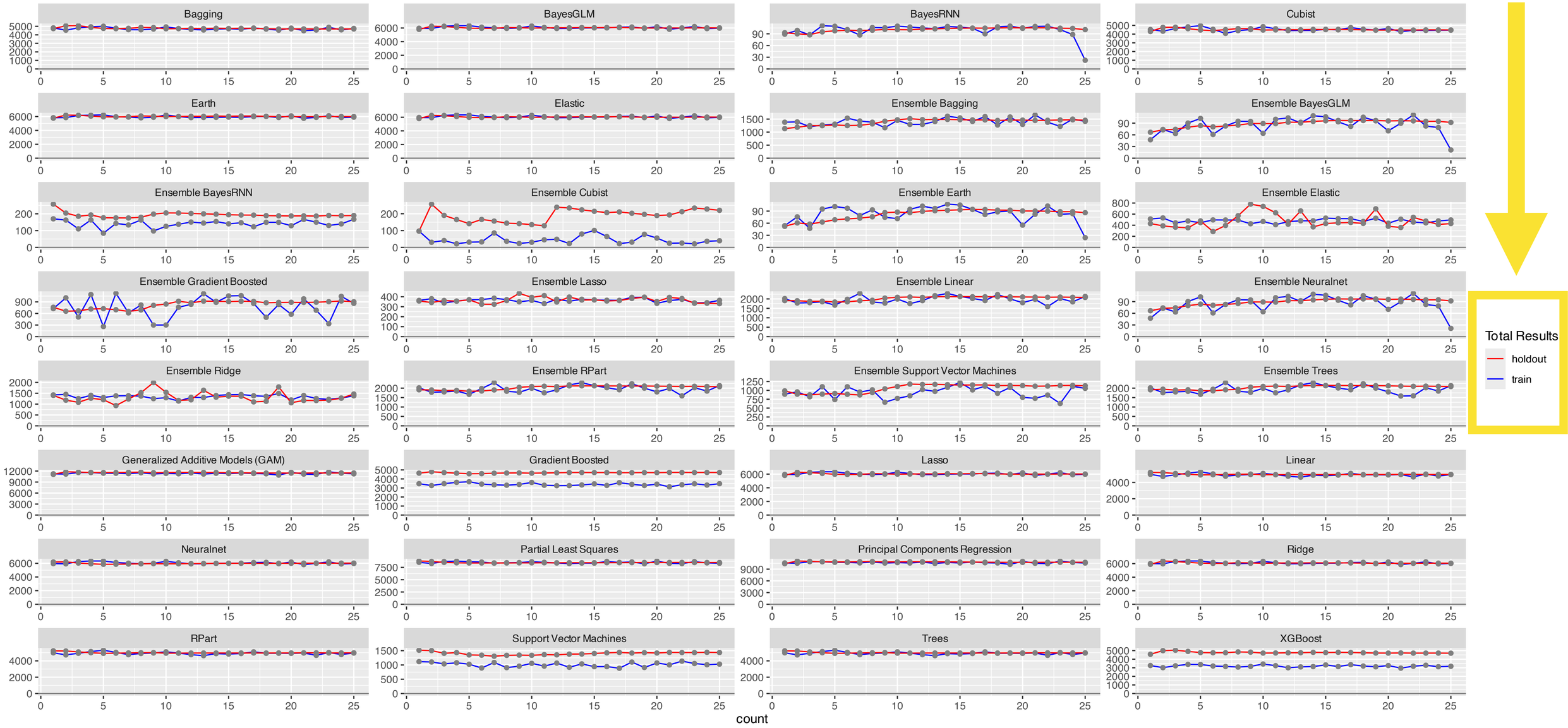
Overfitting by model: Closer to 1 is better.

$$\text{Overfitting} = \frac{\text{RMSE Holdout}}{\text{RMSE Train}}$$




Ensemble Earth
1.00
0.95
0.90
0.85
0.80
0
5
10
15
20
25
Ensemble Neuralnet
1.4
1.3
1.2
1.1
1.0
0
5
10
15
20
25

Overfitting (Train vs holdout) results by resample and model. Free scales
Root Mean Squared Error by model, lower is better.
The black horizontal line is 0.



Separators:

What are the top and bottom parts of the target data doing that nothing else is doing?

Don’t study the top performers. Study what the top (and bottom) performers are doing that no one else is doing: Study what *separates* the top results from the rest of the results. That’s gold.

How to create Separators (automatically part of NumericEnsembles)

Sort the target column from highest to lowest

Create two data sets: The highest 5% and the lowest 5% (of the target column)

Plot the highest and lowest 5% data sets against the other features in the original data set

This will expose problems, risk, and much more. It will provoke questions and (hopefully) paths to better solutions

NumericEnsembles does all of this automatically

Demo: The Insurance data set

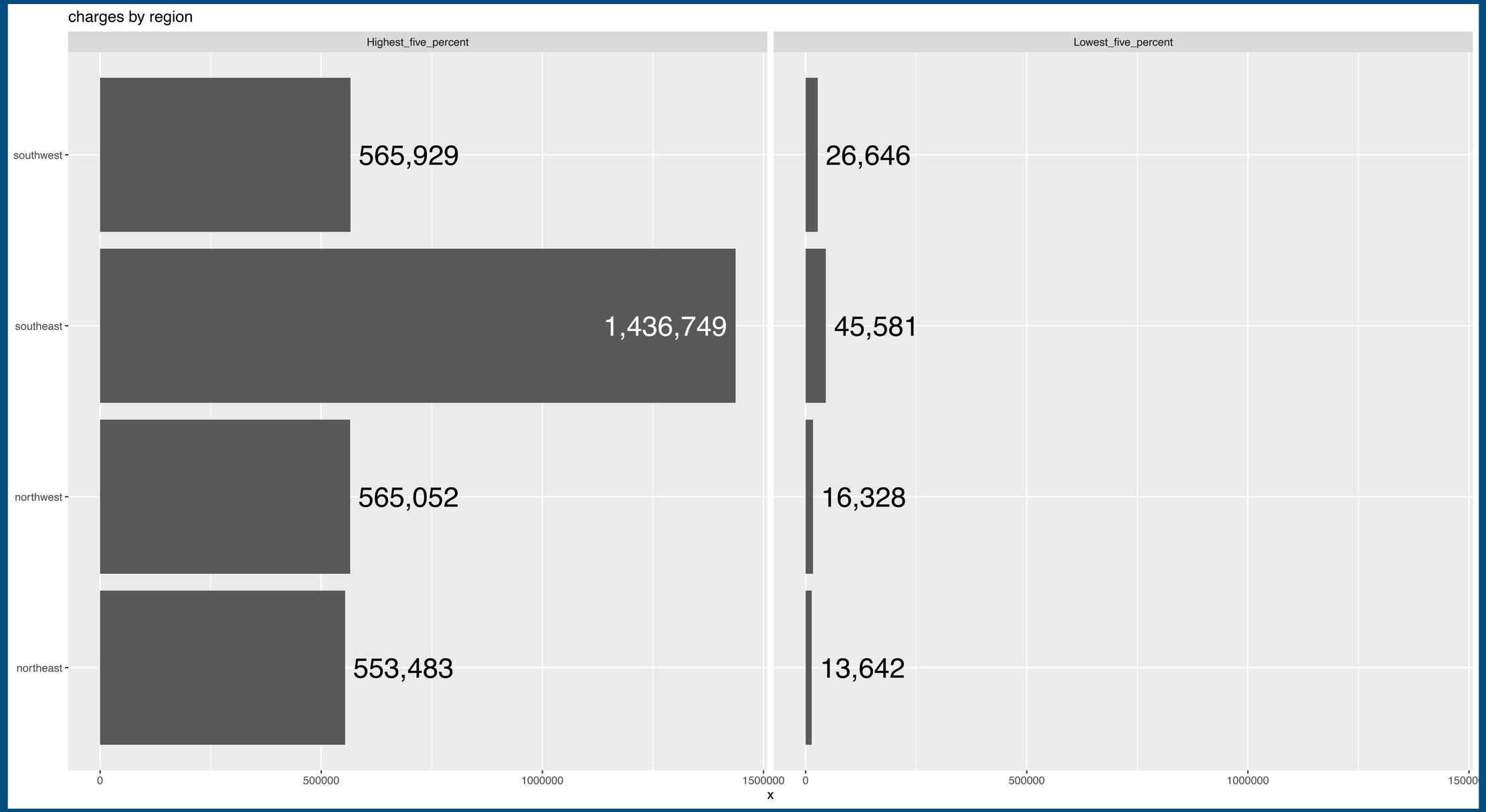
Highest 20 rows and lowest 20 rows by charges

(NumericEnsembles automatically does top 5% and bottom 5%)

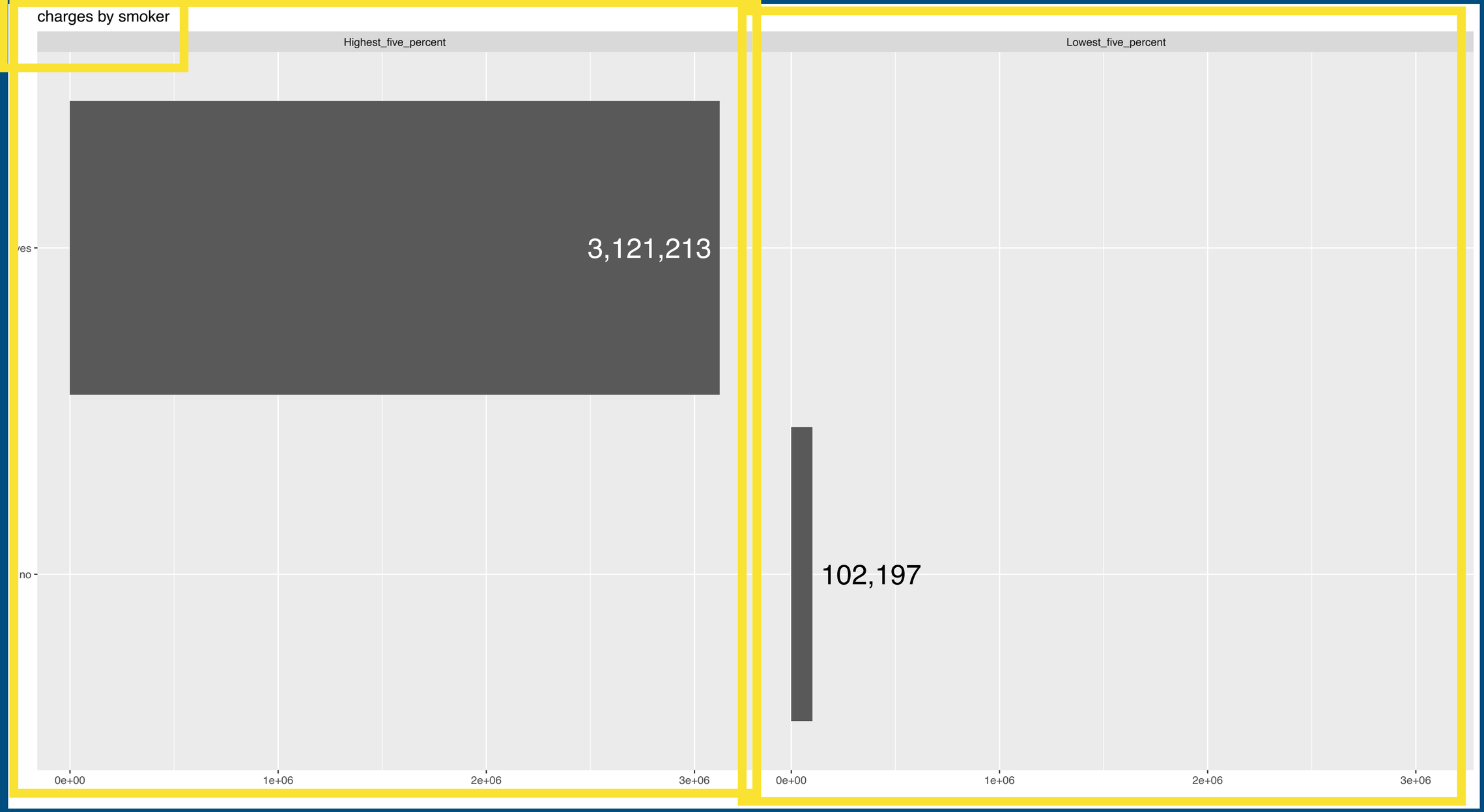
	age	sex	bmi	children	smoker	region	charges
1	54	female	47.410	0	yes	southeast	63770.43
2	45	male	30.360	0	yes	southeast	62592.87
3	52	male	34.485	3	yes	northwest	60021.40
4	31	female	38.095	1	yes	northeast	58571.07
5	33	female	35.530	0	yes	northwest	55135.40
6	60	male	32.800	0	yes	southwest	52590.83
7	28	male	36.400	1	yes	southwest	51194.56
8	64	male	36.960	2	yes	southeast	49577.66
9	59	male	41.140	1	yes	southeast	48970.25
10	44	female	38.060	0	yes	southeast	48885.14
11	63	female	37.700	0	yes	southwest	48824.45
12	57	male	42.130	1	yes	southeast	48675.52
13	60	male	40.920	0	yes	southeast	48673.56
14	54	male	40.565	3	yes	northeast	48549.18
15	61	female	36.385	1	yes	northeast	48517.56
16	60	male	39.900	0	yes	southwest	48173.36
17	64	female	33.800	1	yes	southwest	47928.03
18	59	female	36.765	1	yes	northeast	47896.79
19	58	male	36.955	2	yes	northwest	47496.49
20	51	male	42.900	2	yes	southeast	47462.89

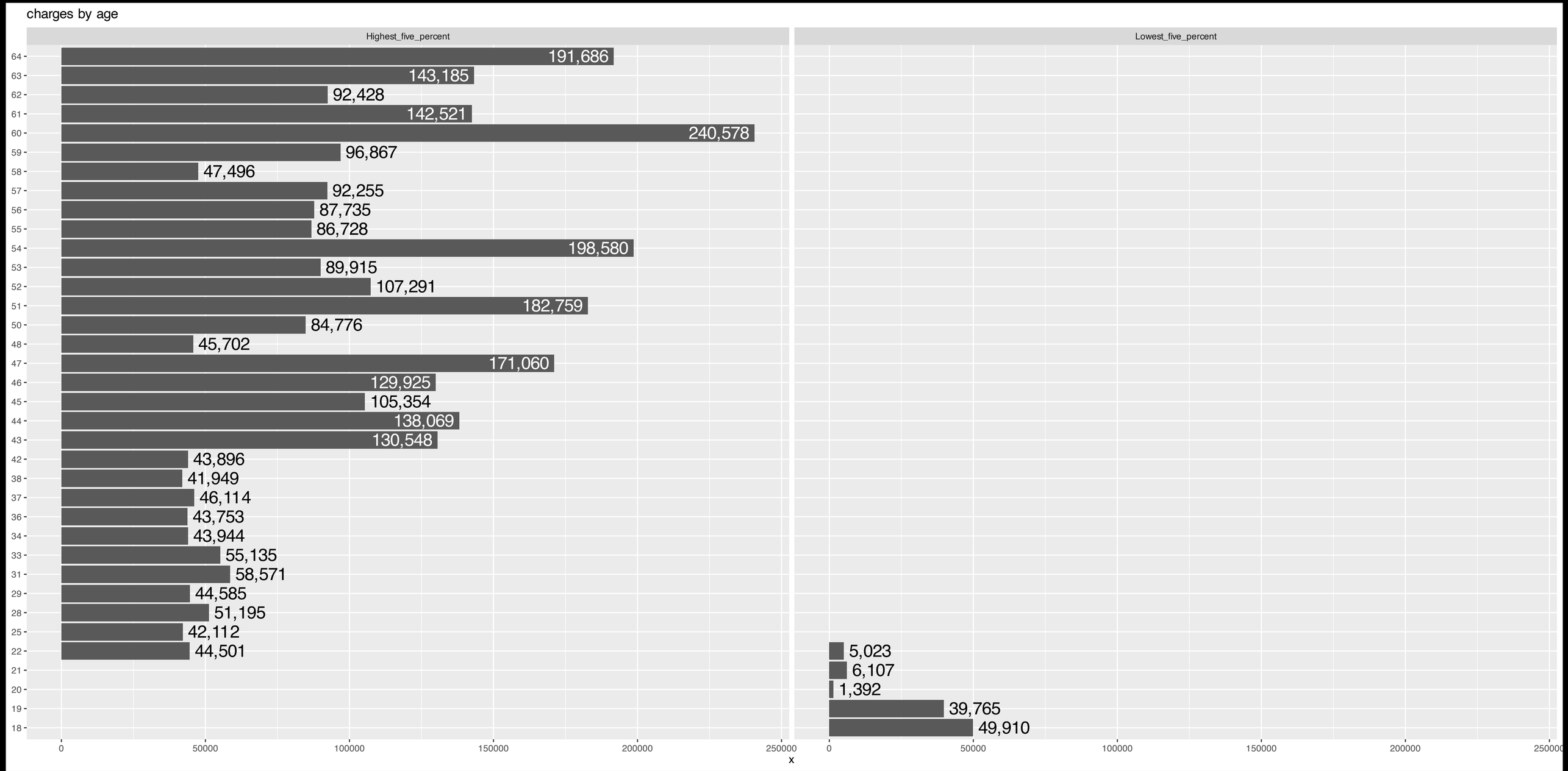
	age	sex	bmi	children	smoker	region	charges
1319	20	male	33.33	0	no	southeast	1391.529
1320	19	male	35.40	0	no	southwest	1263.249
1321	19	male	34.40	0	no	southwest	1261.859
1322	19	male	34.10	0	no	southwest	1261.442
1323	19	male	30.40	0	no	southwest	1256.299
1324	19	male	28.70	0	no	southwest	1253.936
1325	19	male	27.60	0	no	southwest	1252.407
1326	19	male	20.70	0	no	southwest	1242.816
1327	19	male	20.30	0	no	southwest	1242.260
1328	19	male	19.80	0	no	southwest	1241.565
1329	18	male	53.13	0	no	southeast	1163.463
1330	18	male	43.01	0	no	southeast	1149.396
1331	18	male	41.14	0	no	southeast	1146.797
1332	18	male	37.29	0	no	southeast	1141.445
1333	18	male	34.43	0	no	southeast	1137.470
1334	18	male	34.10	0	no	southeast	1137.011
1335	18	male	33.66	0	no	southeast	1136.399
1336	18	male	33.33	0	no	southeast	1135.941
1337	18	male	30.14	0	no	southeast	1131.507
1338	18	male	23.21	0	no	southeast	1121.874

Separator: This one does not separate anything useful



Separator: This is an excellent example

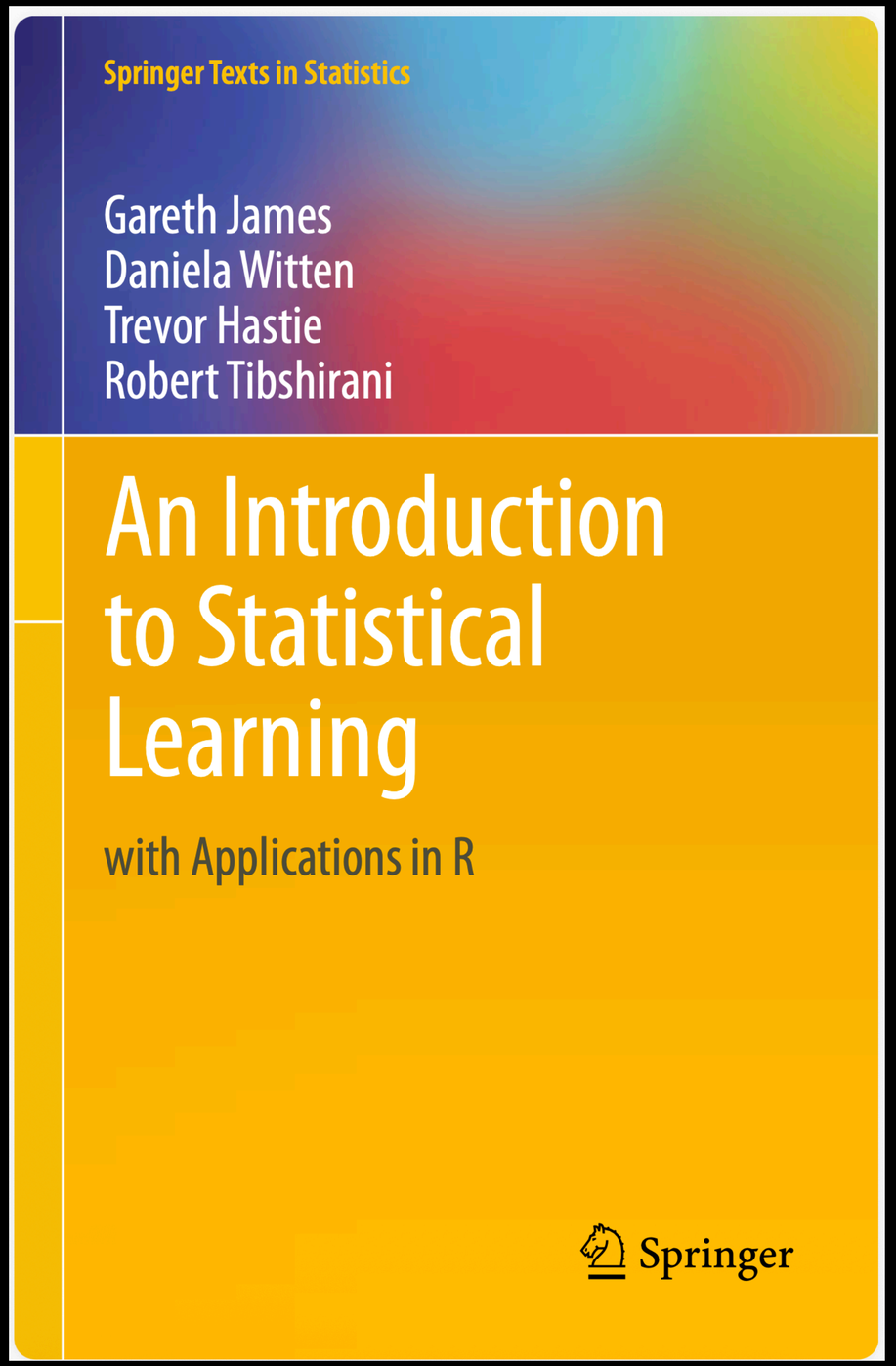



charges by age
Highest_five_percent
Lowest_five_percent
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
48
47
46
45
44
43
42
38
37
36
34
33
31
29
28
25
22
21
20
19
18
191,686
143,185
92,428
142,521
240,578
96,867
47,496
92,255
87,735
86,728
198,580
89,915
107,291
182,759
84,776
45,702
171,060
129,925
105,354
138,069
130,548
43,896
41,949
46,114
43,753
43,944
55,135
58,571
44,585
51,195
42,112
44,501
5,023
6,107
1,392
39,765
49,910
0
50000
100000
150000
200000
250000
x

3. LogisticEnsembles

LogisticEnsembles

The key insight/ observation to LogisticEnsembles from ISLR (page 163)



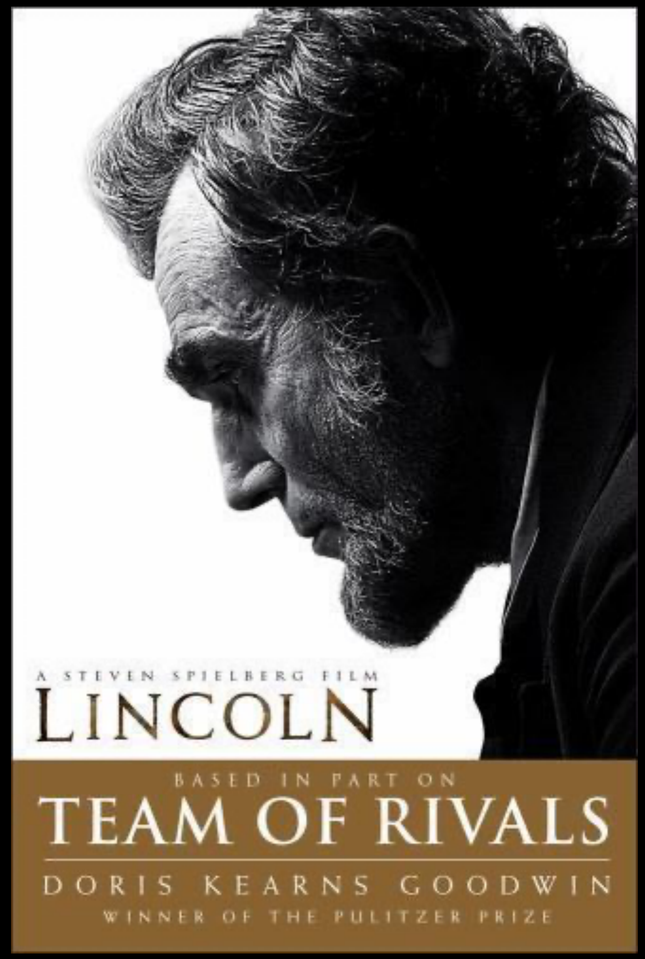
```
> library(class)
> train.X=cbind(Lag1,Lag2)[train,]
> test.X=cbind(Lag1,Lag2)[!train,]
> train.Direction=Direction[train]
```

Now the `knn()` function can be used to predict the market's movement for the dates in 2005. We set a random seed before we apply `knn()` because if several observations are tied as nearest neighbors, then `R` will randomly break the tie. Therefore, a seed must be set in order to ensure reproducibility of results.

```
> set.seed(1)
> knn.pred=knn(train.X,test.X,train.Direction,k=1)
> table(knn.pred,Direction.2005)
        Direction.2005
knn.pred Down Up
    Down   43 58
    Up     68 83
> (83+43)/252
[1] 0.5
```

LogisticEnsembles: The list of 24 logistic models

Model	Individual	Ensemble
Bagging	✗	✓
BayesGLM	✓	✗
BayesRNN	✓	✗
C50	✓	✓
Cubist	✓	✗
Flexible Discriminant Analysis	✓	✗
Generalized Additive Models	✓	✗
Generalized Linear Models	✓	✗
Gradient Boosted	✗	✓
Linear	✓	✗
Linear Discriminant Analysis	✓	✗
Partial Least Squares	✗	✓
Penalized Discriminant Analysis	✓	✓
Quadratic Discriminant Analysis	✓	✗
Random Forest	✓	✗
Part	✗	✓
Support Vector Machines	✓	✓
Trees	✓	✓
XGBoost	✓	✓
Total	15	9



LogisticEnsembles Example: The Pima Indians diabetes data set

768 rows x 9 columns

0	1
500	268

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0

Demo of LogisticEnsembles

RStudio

Go to file/function | Addins | Project: (None)

Diabetes example.R

Source on Save | Run | Source

```
1  library(rgl)
2  library(LogisticEnsembles)
3
4  start_time <- Sys.time()
5  Logistic(data = Diabetes,
6           colnum = 9,
7           numresamples = 25,
8           remove_VIF_greater_than <- 5.00,
9           save_all_trained_models = "N",
10          save_all_plots = "N",
11          set_seed = "N",
12          how_to_handle_strings = 1,
13          do_you_have_new_data = "N",
14          remove_data_correlations_greater_than = 1.00,
15          remove_ensemble_correlations_greater_than = 0.99,
16          use_parallel = "Y",
17          train_amount = 0.50,
18          test_amount = 0.25,
19          validation_amount = 0.25)
20 end_time <- Sys.time()
21 duration <- end_time - start_time
22 duration
23 warnings()
```

3:1 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.5.2 · ~/Library/Mobile Documents/com~apple~CloudDocs/Documents/Machine Learning templates in R/

```
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Environment | History | Connections | Tutorial

Files | Plots | Packages | Help | Viewer | Presentation

4. ClassificationEnsembles

```
> head(Cleveland_heart)
  Age  Sex Chest_pain_type Resting_blood_pressure Cholesteral Fasting_blood_sugar
1  67 male          asympt                   160         286                  fal
2  67 male          asympt                   120         229                  fal
3  37 male          notang                   130         250                  fal
4  41  fem          abnang                   130         204                  fal
5  56 male          abnang                   120         236                  fal
6  62  fem          asympt                   140         268                  fal
  Max_heart_rate Exercise_induced_angina Old_peak Slope Sick_or_buff Class
1            108                   true      1.5  flat         sick    S2
2            129                   true      2.6  flat         sick    S1
3            187                    fal      3.5  down         buff     H
4            172                    fal      1.4    up         buff     H
5            178                    fal      0.8    up         buff     H
6            160                    fal      3.6  down         sick    S3
```

RStudio

Go to file/function | Addins | Project: (None)

Cleveland Heart for demo.R

Source on Save | Run | Source

```
1  library(ClassificationEnsembles)
2
3  start_time <- Sys.time()
4  Classification(data = Cleveland_heart,
5                 colnum = 12,
6                 numresamples = 25,
7                 predict_on_new_data = "N",
8                 save_all_plots = "N",
9                 how_to_handle_strings = 1,
10                remove_VIF_above <- 5.00,
11                set_seed = "N",
12                save_all_trained_models = "N",
13                scale_all_numeric_predictors_in_data = "N",
14                use_parallel = "N",
15                train_amount = 0.50,
16                test_amount = 0.25,
17                validation_amount = 0.25)
18
19 end_time <- Sys.time()
20 duration <- end_time - start_time
21 duration
22 warnings()
23
```

23:1 (Top Level) R Script

Console | Terminal | Background Jobs

R 4.5.2 · ~/Library/Mobile Documents/com~apple~CloudDocs/Documents/Machine Learning templ

```
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Environment | History | Connections | Tutorial

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export

No ads, no spam, no trackers.

Your activity is never monitored or tracked at all.

Your data is never saved by the packages.

No LLMs are used, no agents are used, your data is never sent anywhere.

Everything runs from each of the ensemble packages and nothing else is needed or called. Period.

4. Summary/Wrap-up

All source code (including examples) is available at:

GitHub.com/InfiniteCuriosity

ensembles1.com

NumericEnsembles.com
ClassificationEnsembles.com
LogisticEnsembles.com
ForecastingEnsembles.com

russconte@mac.com

NumericEnsembles (on CRAN)
ClassificationEnsembles (on CRAN)
LogisticEnsembles (on CRAN)
ForecastingEnsembles (on CRAN)