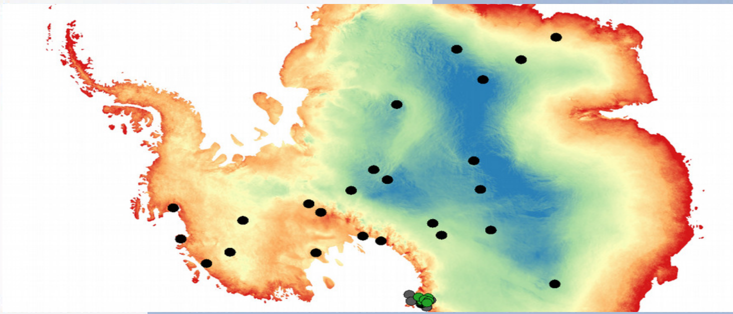
Machine-learning based modelling of spatial and spatio-temporal environmental data

Part 1: Introduction to:

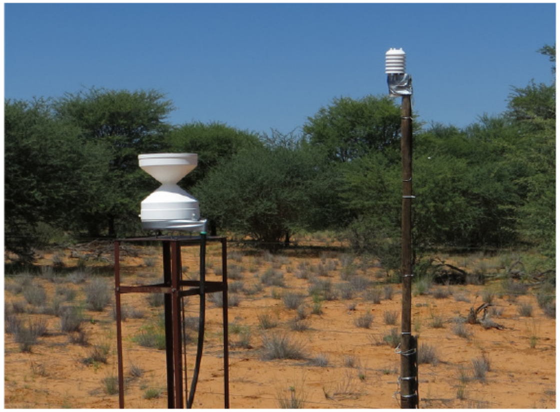
- *Spatio-temporal model training and prediction*
- *Cross validation*
- *Variable selection for spatio-temporal models*

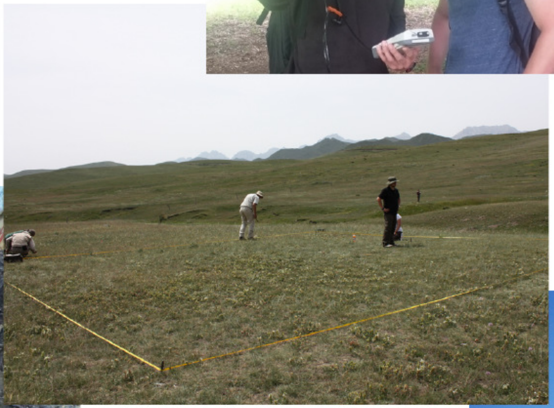


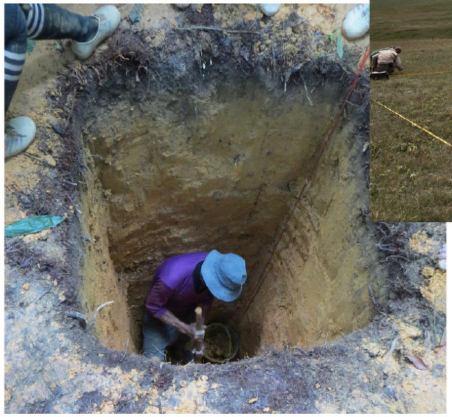
Hanna Meyer

Common problem in environmental science

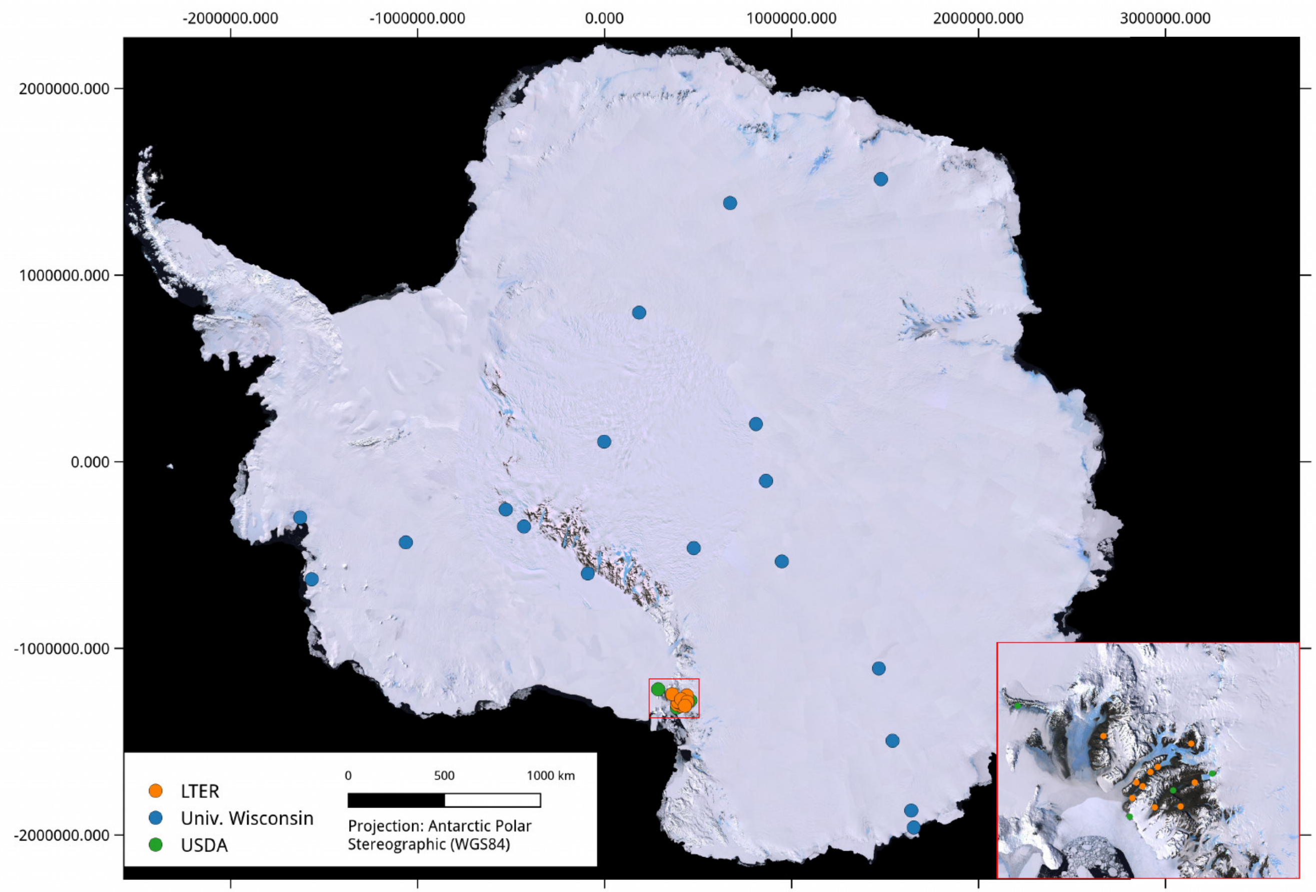
Local sampling vs need for spatially continuous information



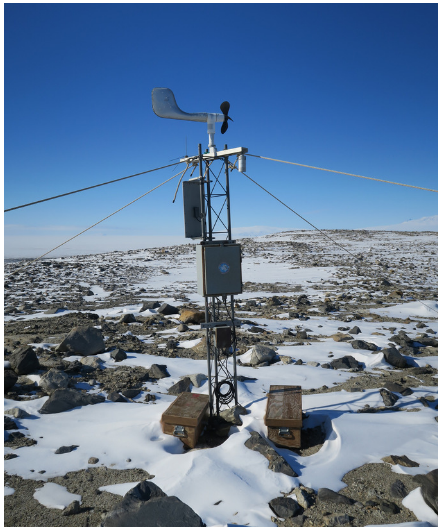




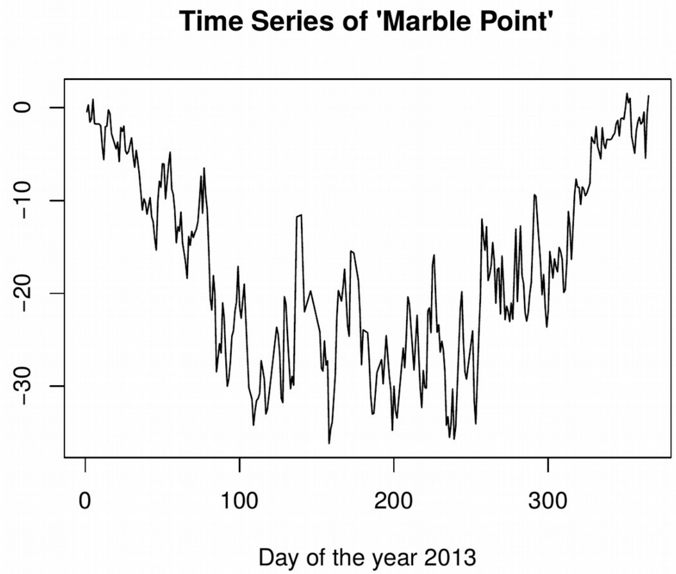
Example: Monitoring Air Temperature in Antarctica



Meyer et al. (2016)

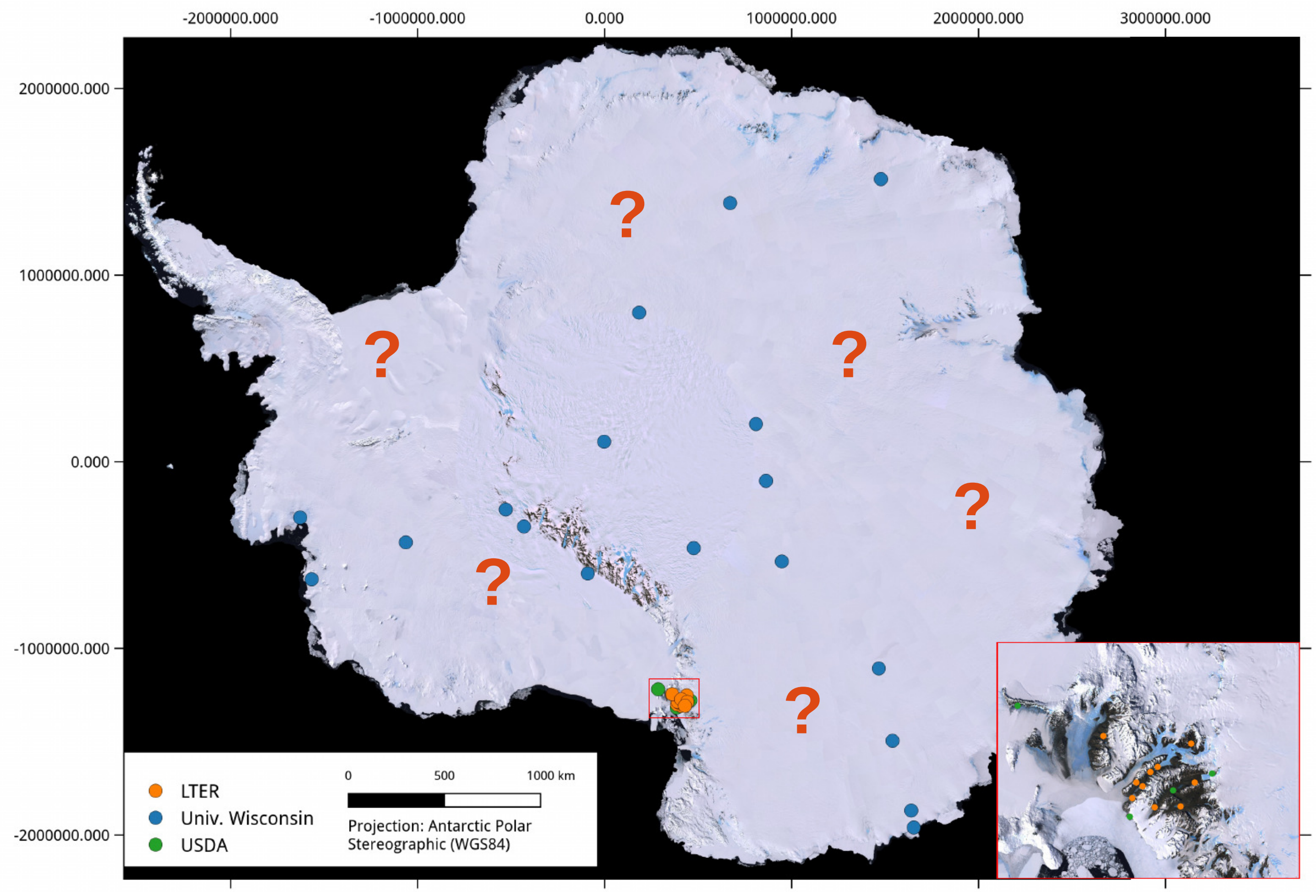


http://amrc.ssec.wisc.edu/

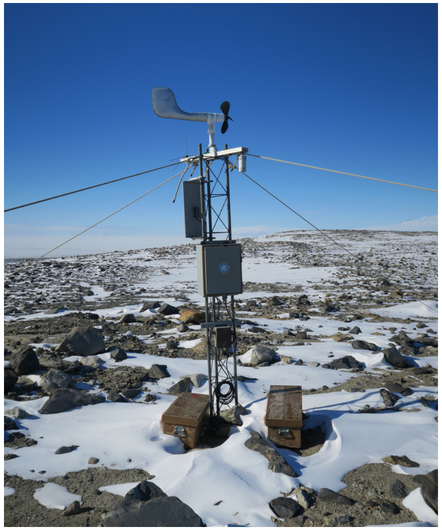


Philipps Universität Marburg

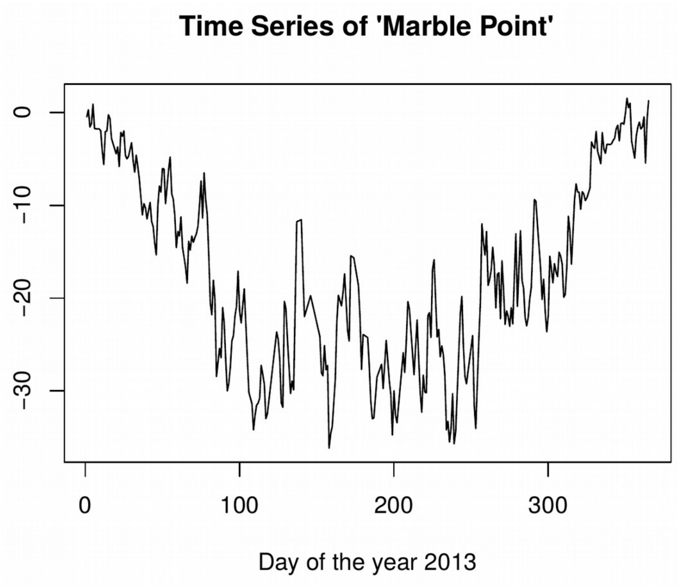
Example: Monitoring Air Temperature in Antarctica



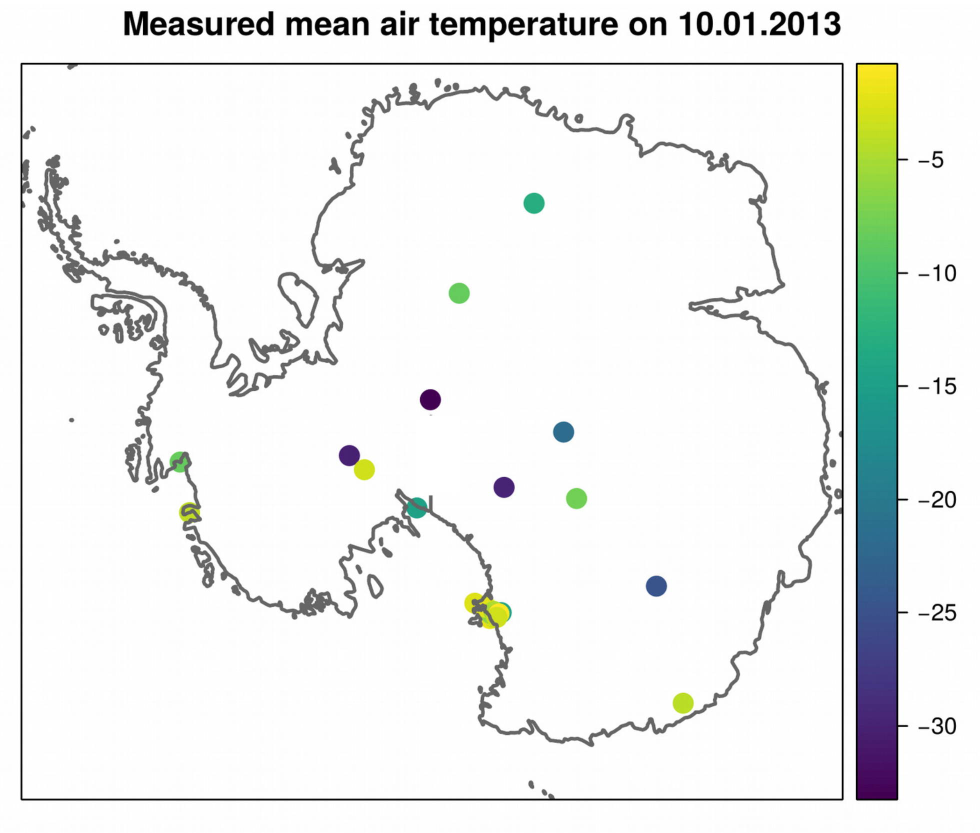
Meyer et al. (2016)

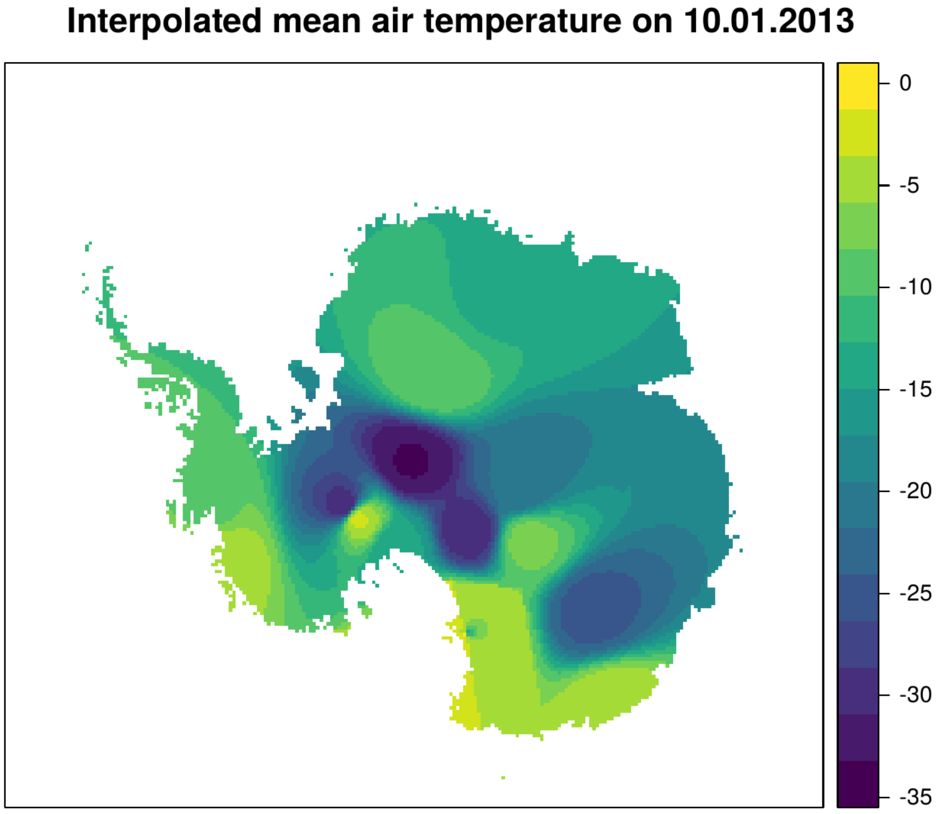


http://amrc.ssec.wisc.edu/



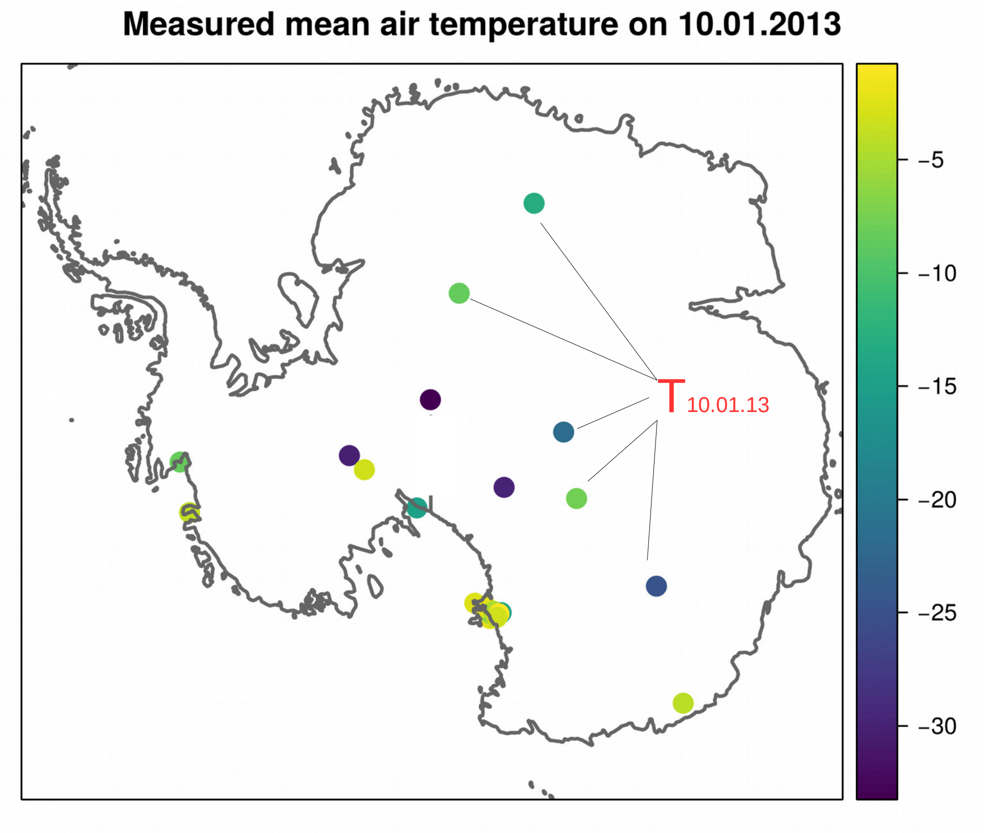
First approach: Simple spatial interpolation

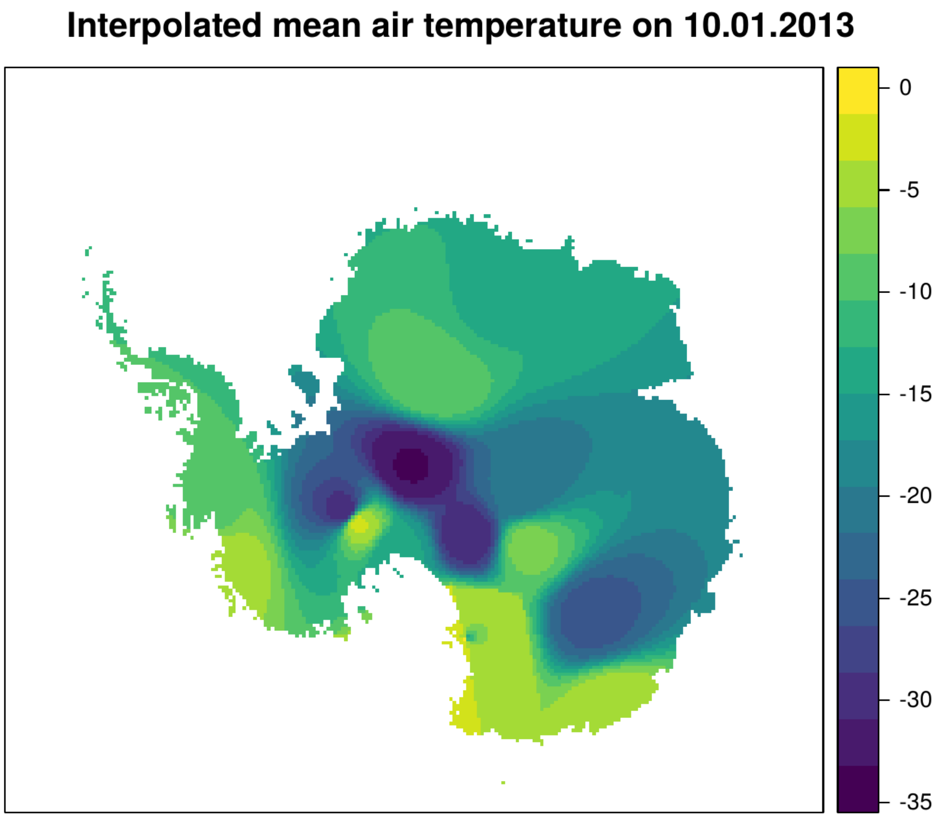




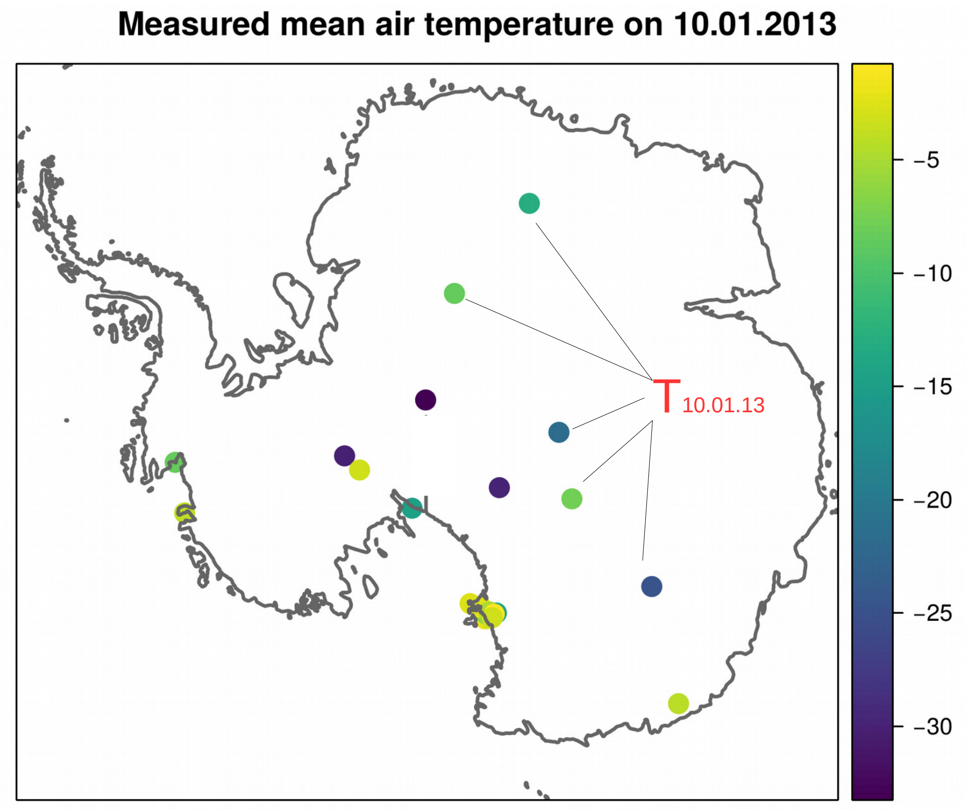
Philipps Universität Marburg

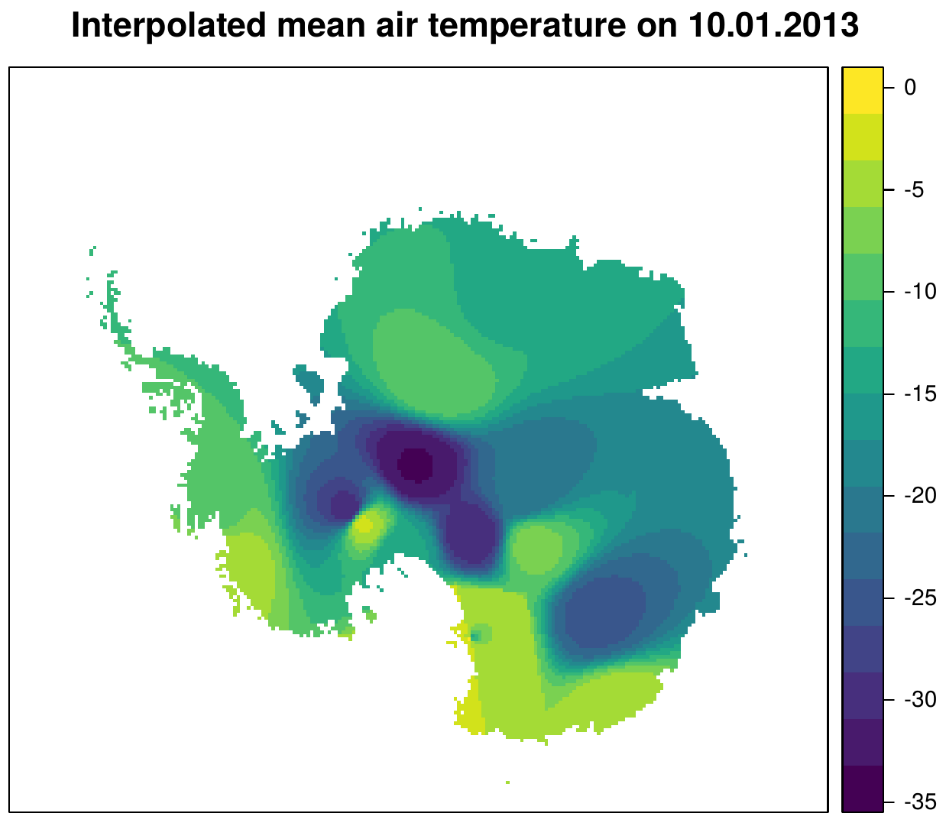
First approach: Simple spatial interpolation





First approach: Simple spatial interpolation

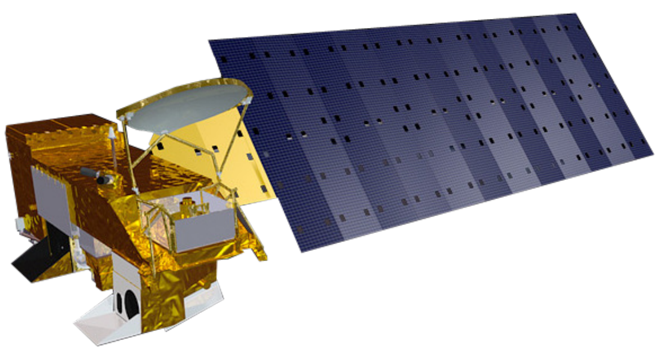




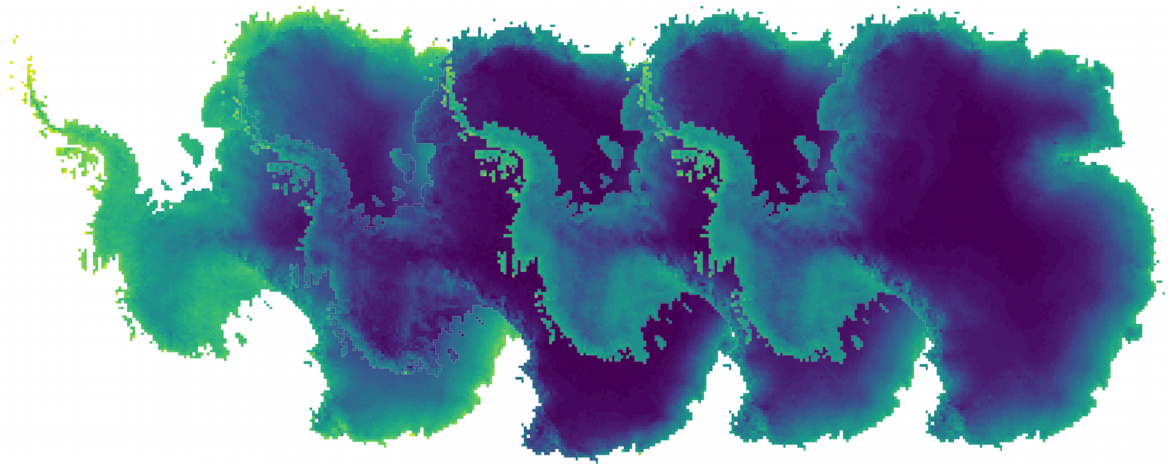
Problem: Distance not a always a meaningful predictor,
Dependency on field data

Spatio-temporal predictors are needed

- No way around remote sensing (e.g. from drones, air planes, satellites)!!
- Assumption: Spectral properties are related to the target
- For this example: e.g. MODIS LST (4 times per day, 1km spatial resolution)

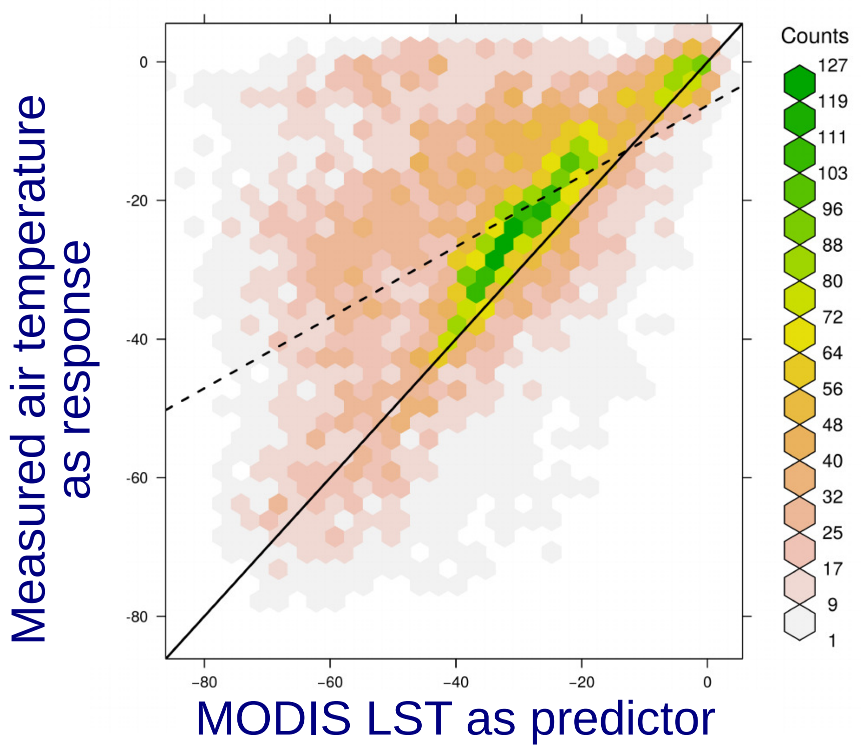


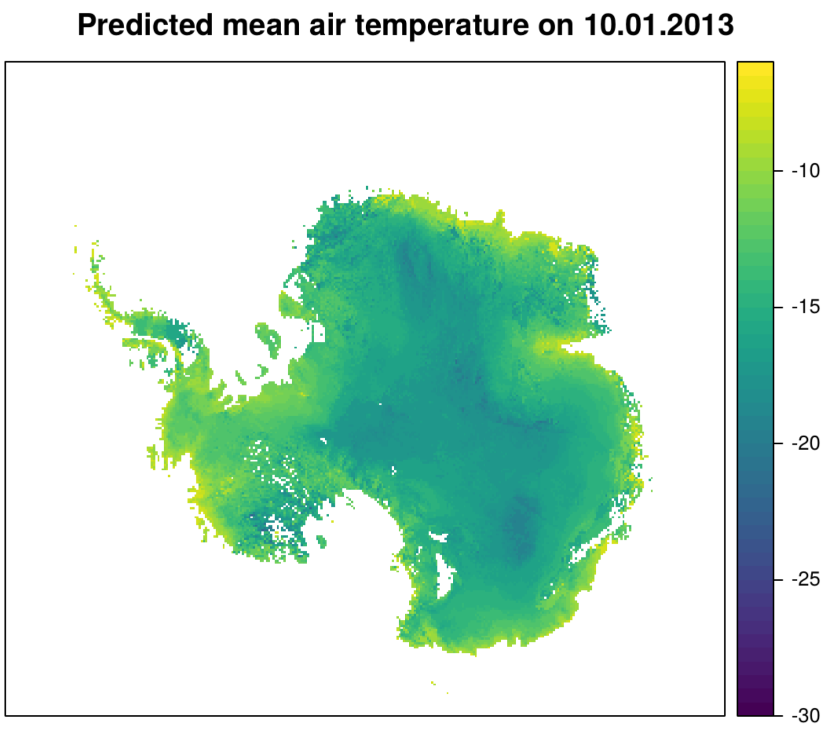
By National Aeronautics and Space Administration (NASA) [Public domain], via Wikimedia Commons



Parametric modelling approaches

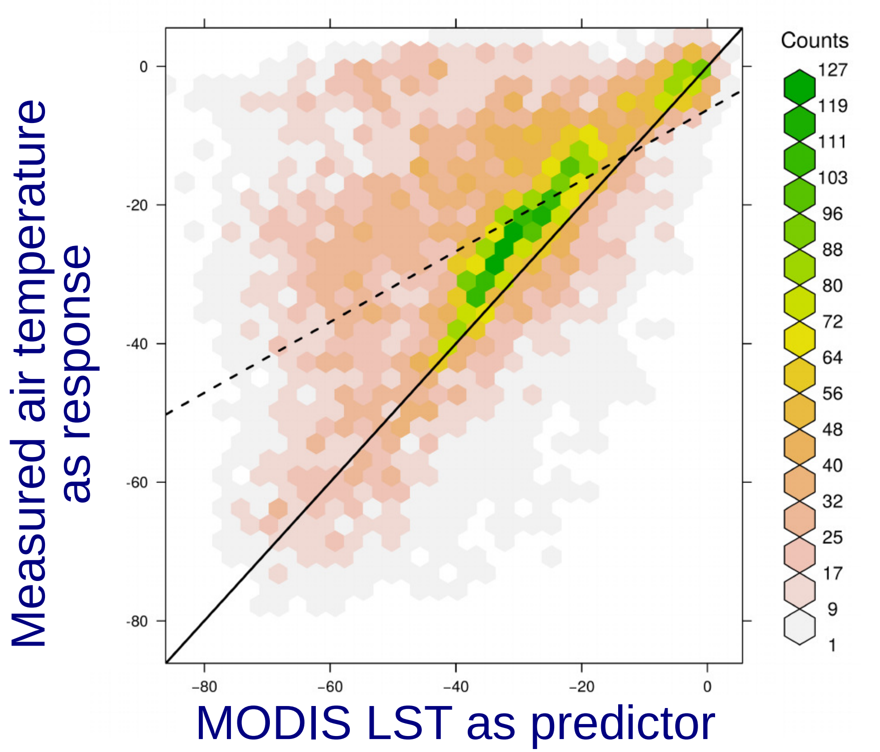
$$T_{air} = -6.26+0.51*LST$$

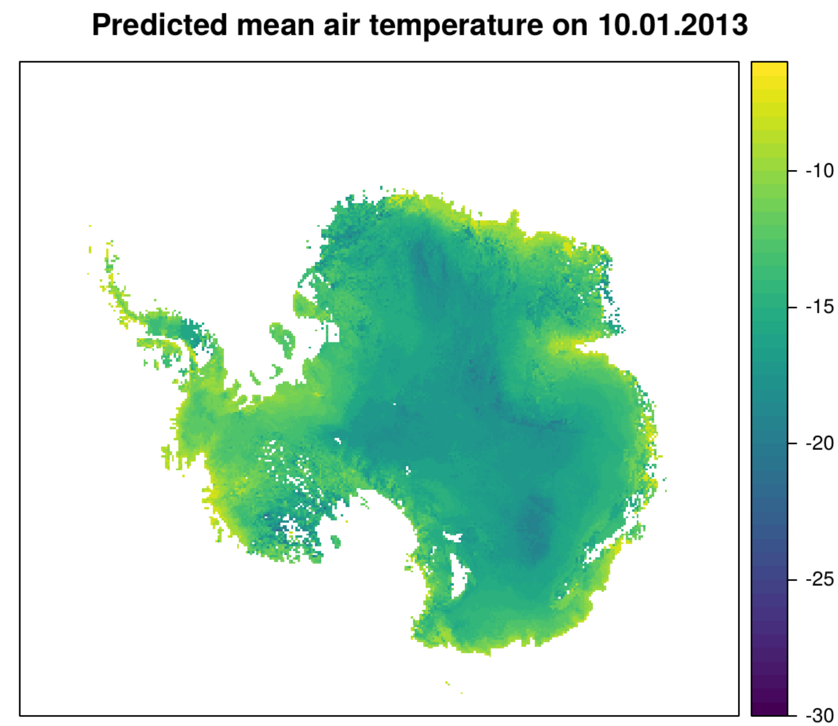




Parametric modelling approaches

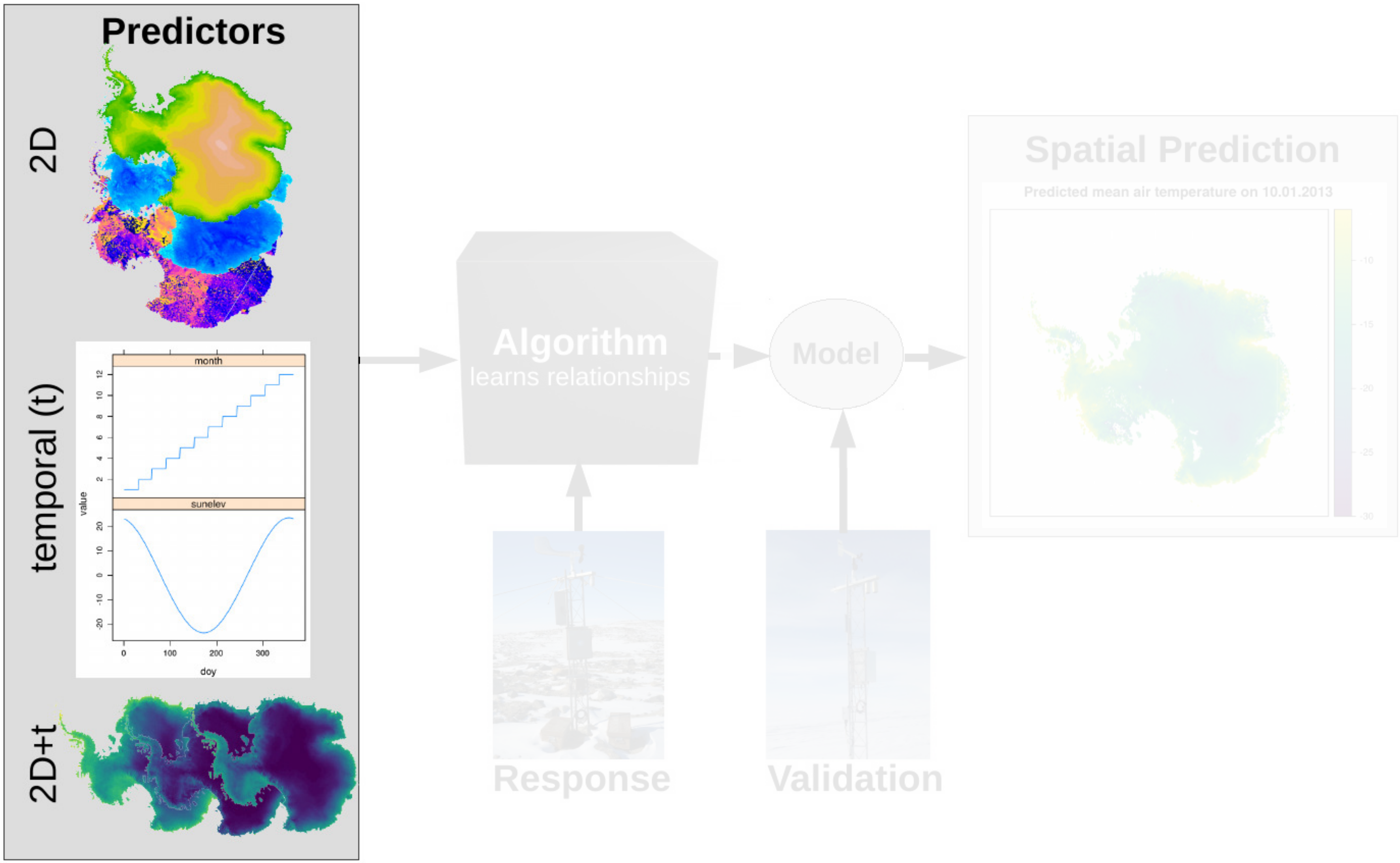
$$T_{air} = -6.26+0.51*LST$$





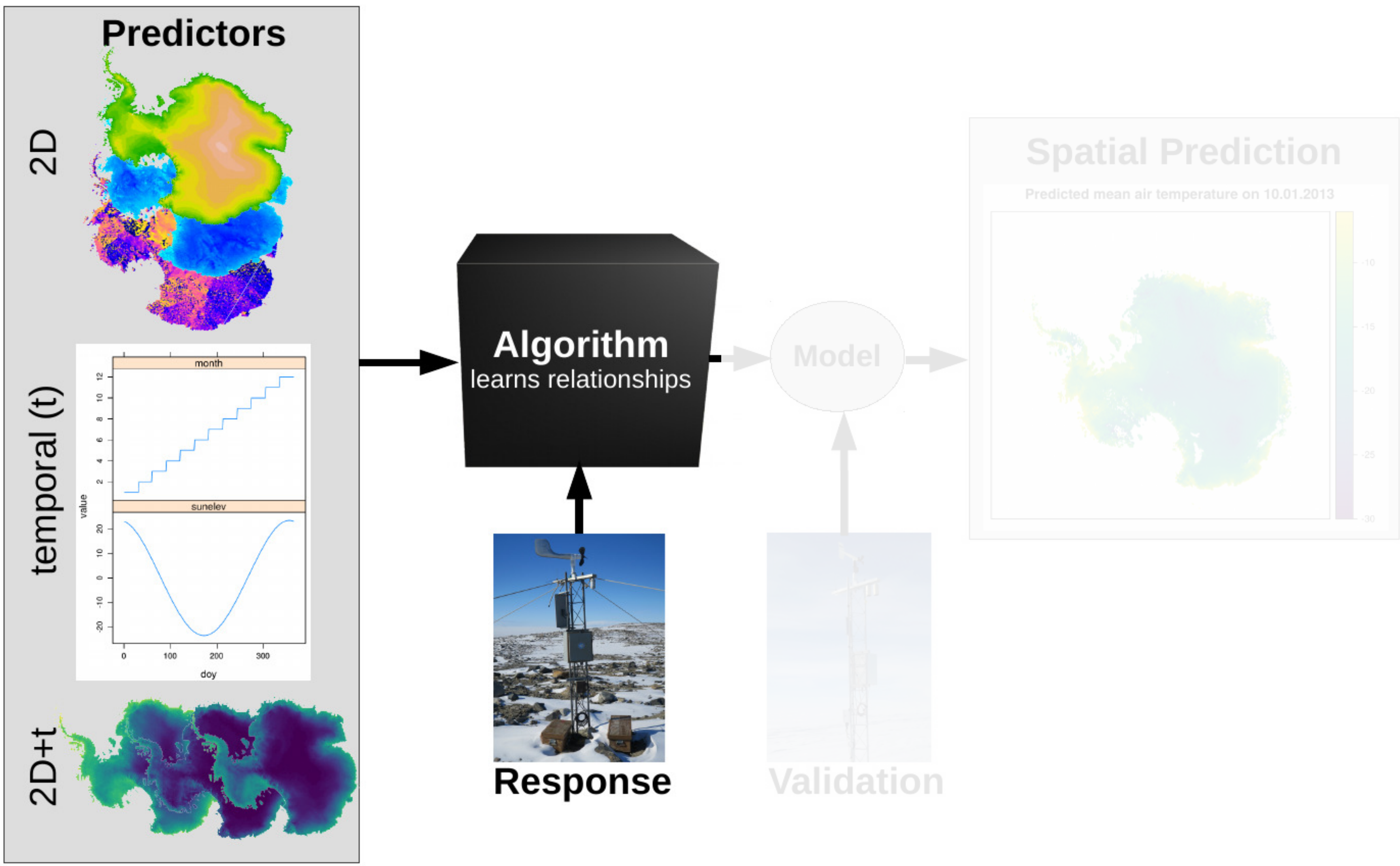
Problem: fixed relationships, limited number of predictors,
But need for flexible models

The Machine learning way

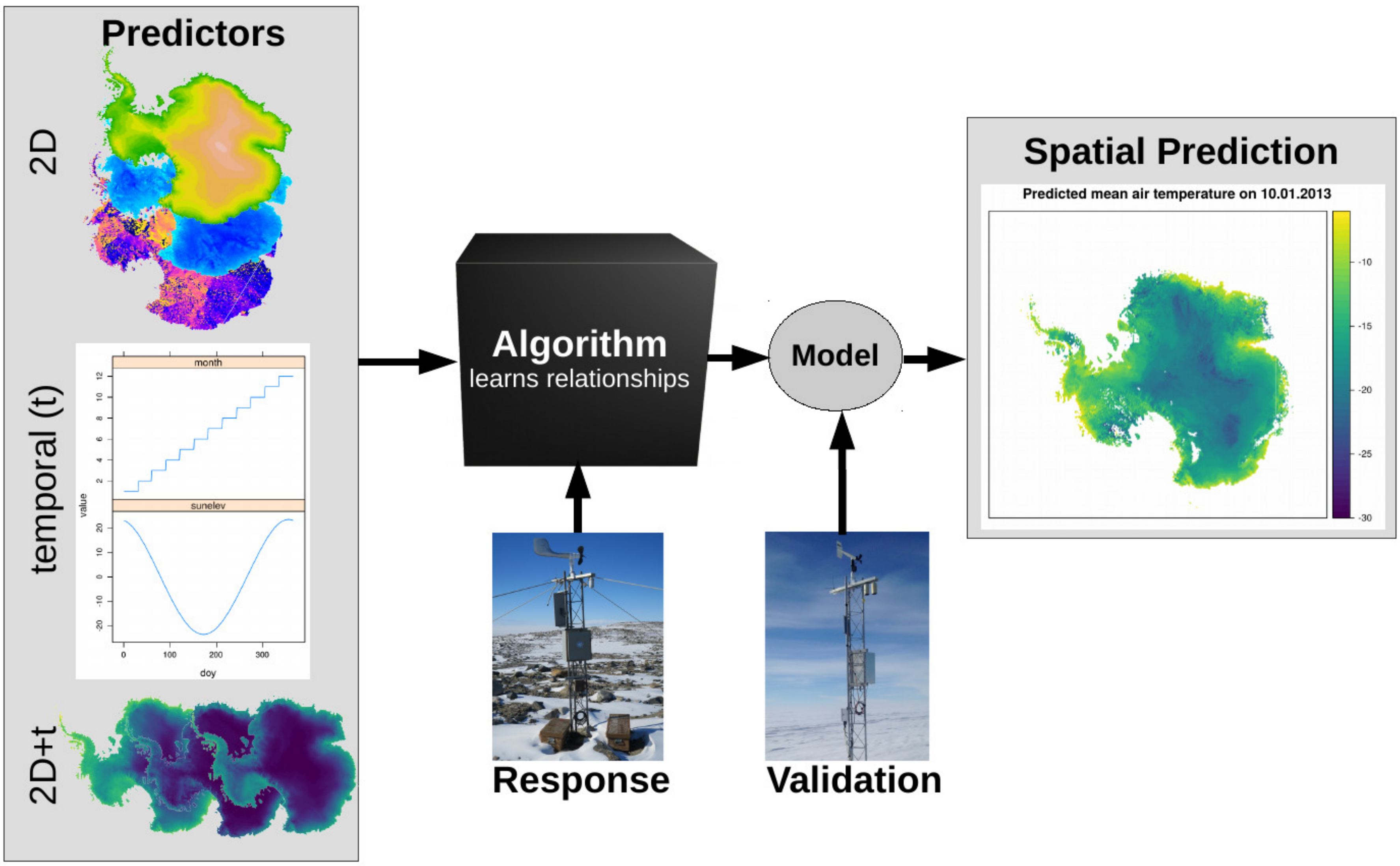


Philipps Universität Marburg

The Machine learning way

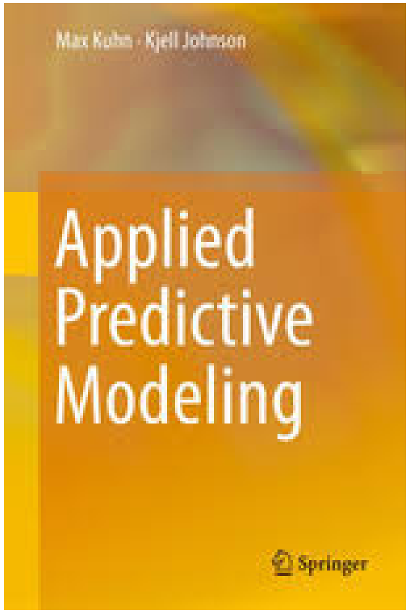


The Machine learning way



Machine learning in R

- Many packages for different ML algorithms (e.g. randomForest)
- For classification and regression problems
- Caret (Classification And REgression Training) is a wrapper package allowing access to many algorithms via a unified syntax
  - Overview of algorithms: http://topepo.github.io/caret/train-models-by-tag.html
  - Supporting functionality for cross-validation etc.
  - Further reading: Applied predictive modelling (with R code examples)



Machine learning in R using caret is easy….
Step one: Model training

Training data:

Predictors Response

Station	Date	LST	Elevation	Aspect	...	Measured Tair
A	2017/01/01	-5	1000	S		-2
B	2017/01/01	0	200	S		-2
C	2017/01/01	-10	3000	E		-5
A	2017/07/01	-40	1000	S		-45
B	2017/07/01	-30	200	S		-30
C	2017/07/01	-60	3000	E		-70
A	2017/10/01	-20	1000	S		-22
B	2017/10/01	-10	200	S		-9
C	2017/10/01	-25	3000	E		-30

Machine learning in R using caret is easy….
Step one: Model training

Training data:

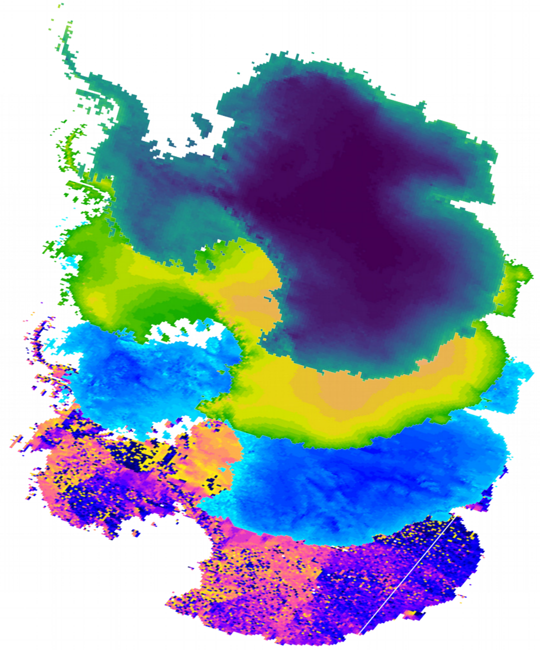
Station	Date	LST	Elevation	Aspect	...	Measured Tair
A	2017/01/01	-5	1000	S		-2
B	2017/01/01	0	200	S		-2
C	2017/01/01	-10	3000	E		-5
A	2017/07/01	-40	1000	S		-45
B	2017/07/01	-30	200	S		-30
C	2017/07/01	-60	3000	E		-70
A	2017/10/01	-20	1000	S		-22
B	2017/10/01	-10	200	S		-9
C	2017/10/01	-25	3000	E		-30

$\underbrace{\text{LST, Elevation, Aspect, ...}}_{\text{Predictors}}$ $\underbrace{\text{Measured Tair}}_{\text{Response}}$

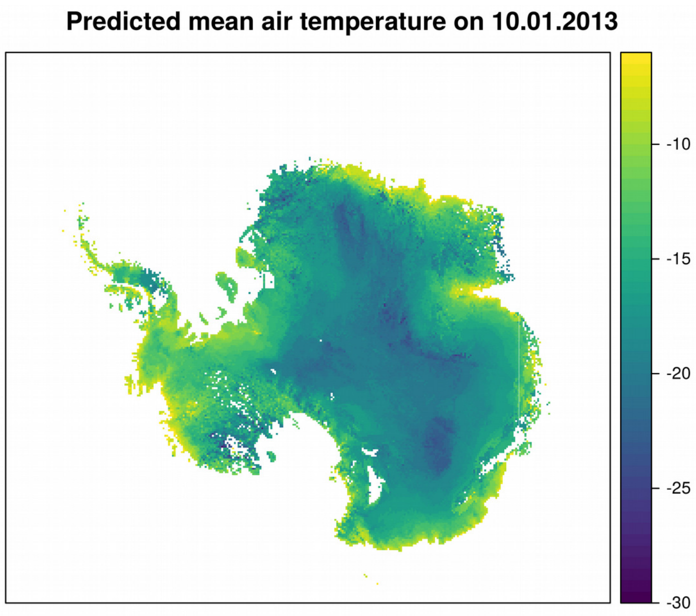
How to do it in R

```
library(caret)
model <- train(predictors,
               response,
               method="rf")
```

Machine learning in R using caret is easy….
Step two: Spatial prediction

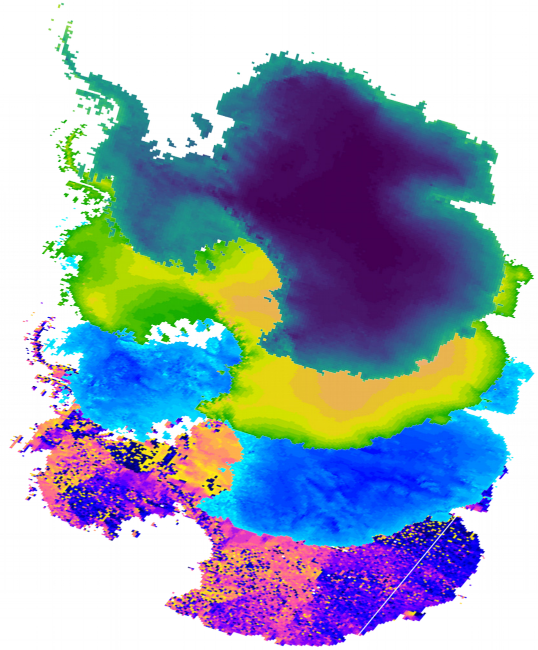


+ trained model =

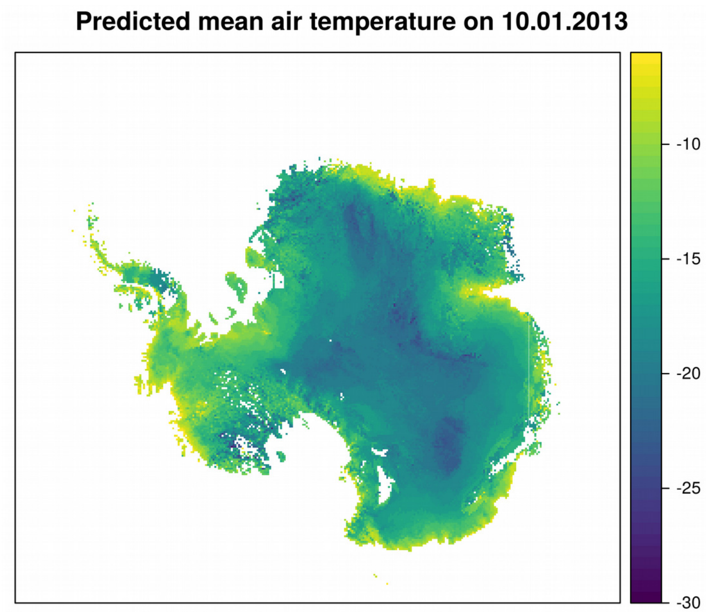


Philipps Universität Marburg

Machine learning in R using caret is easy….
Step two: Spatial prediction



+ trained model =

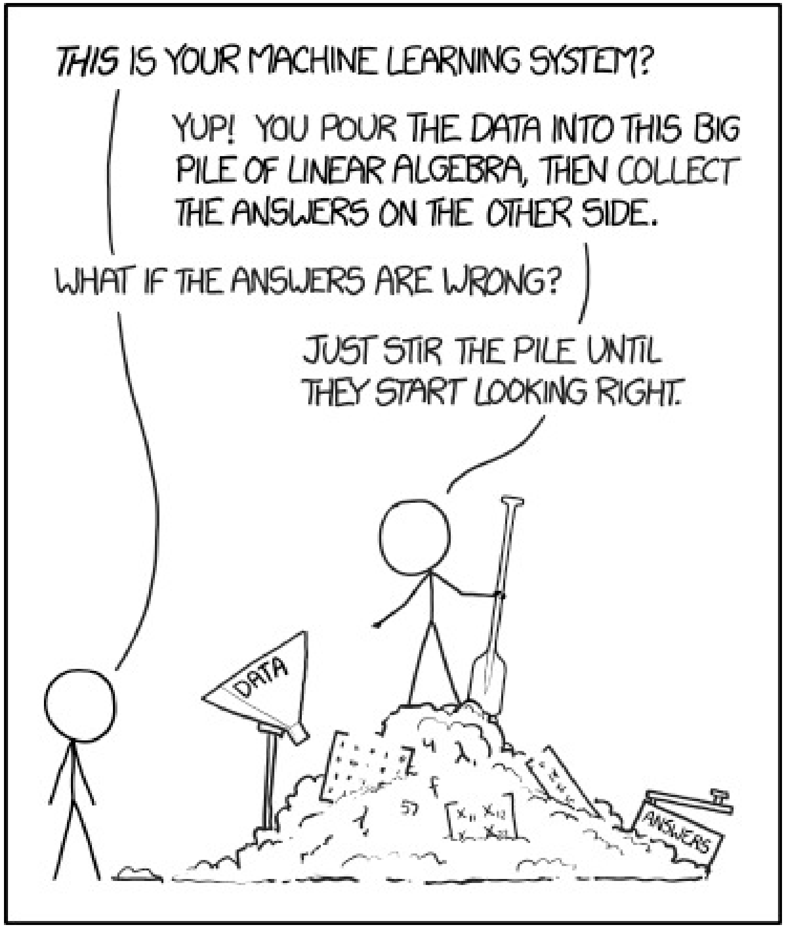


How to do it in R

```
library(raster)
pred_sp <- stack(predictors)
prediction <- predict(pred_sp,model)
```


Philipps Universität Marburg

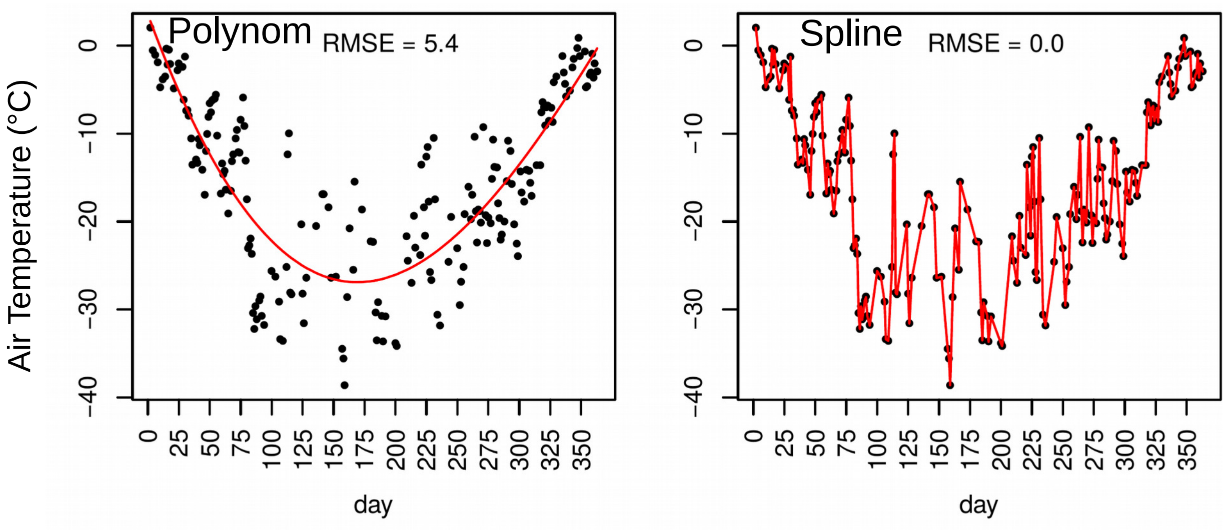
...But how good is the model?



https://xkcd.com/1838/

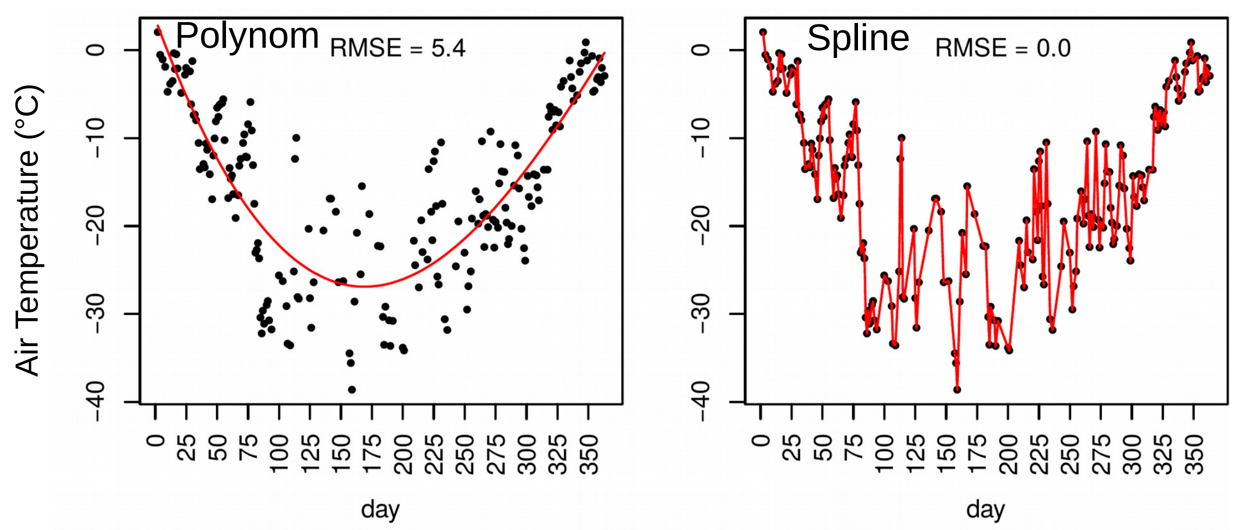
...But how good is the model?

Model training

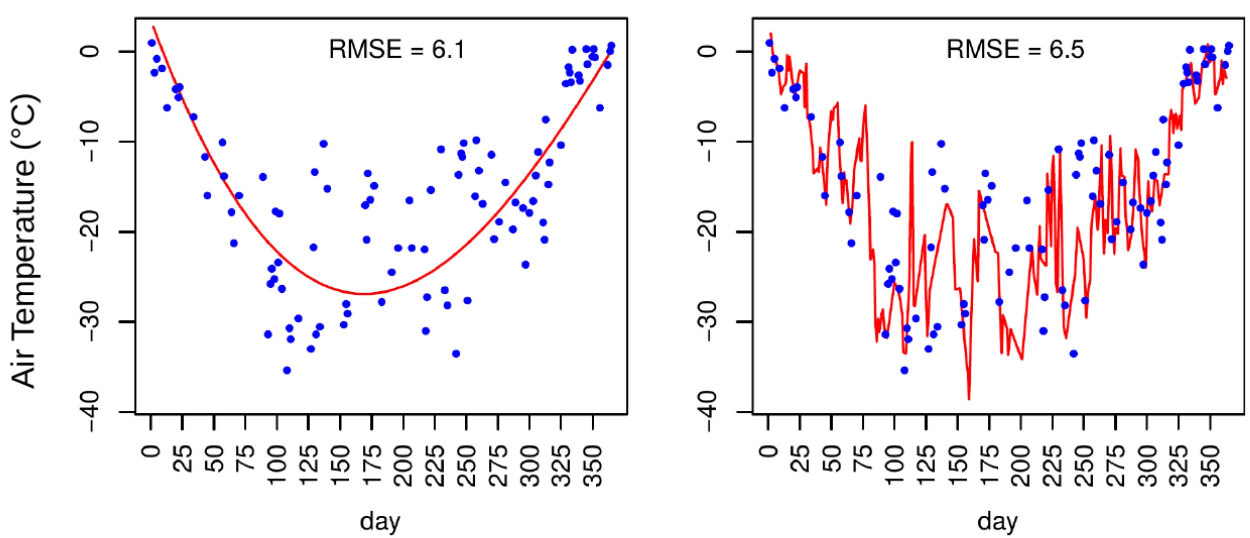


...But how good is the model?

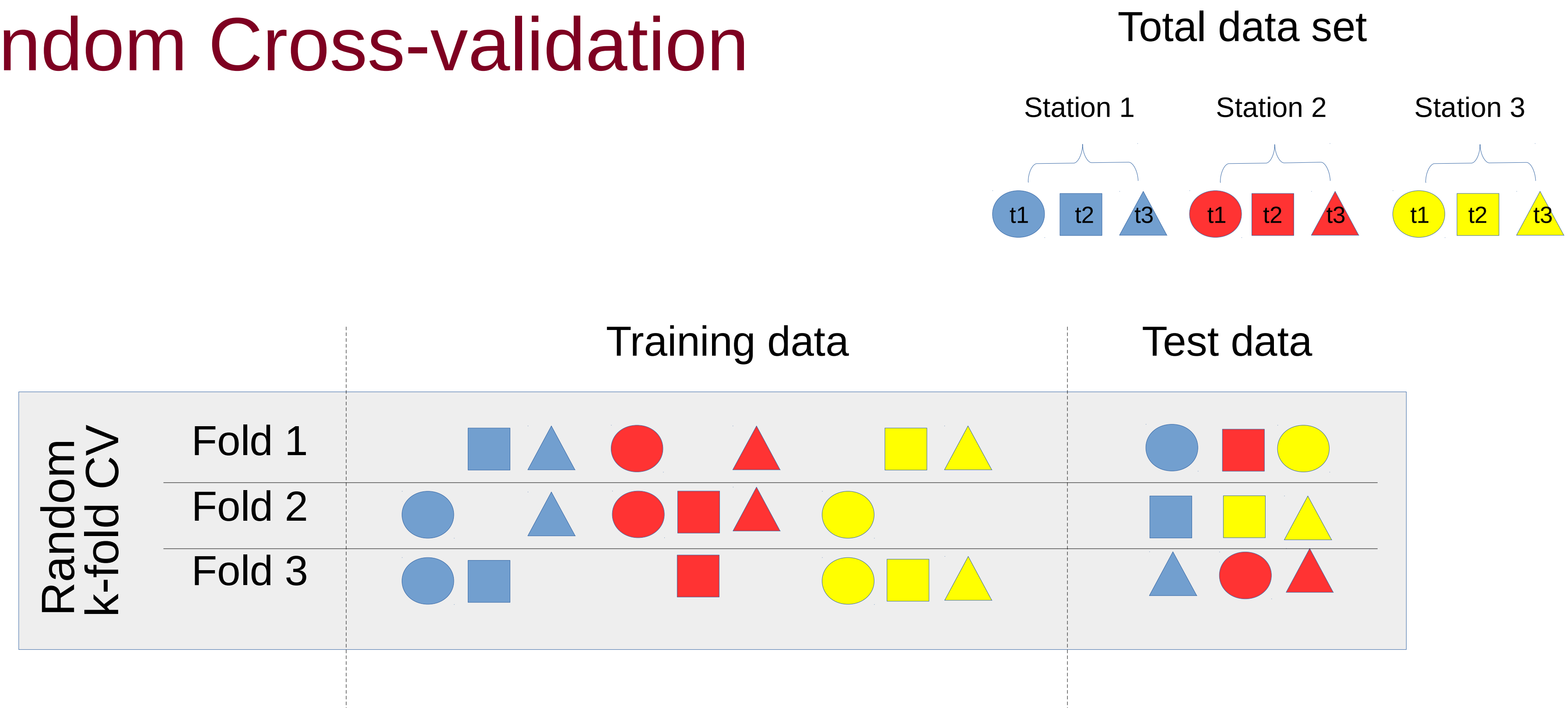
Model training (2/3 of the data)



Model validation (1/3 of the data)



Random Cross-validation



Meyer et al. (2018)

Random Cross-validation

How to do it in R

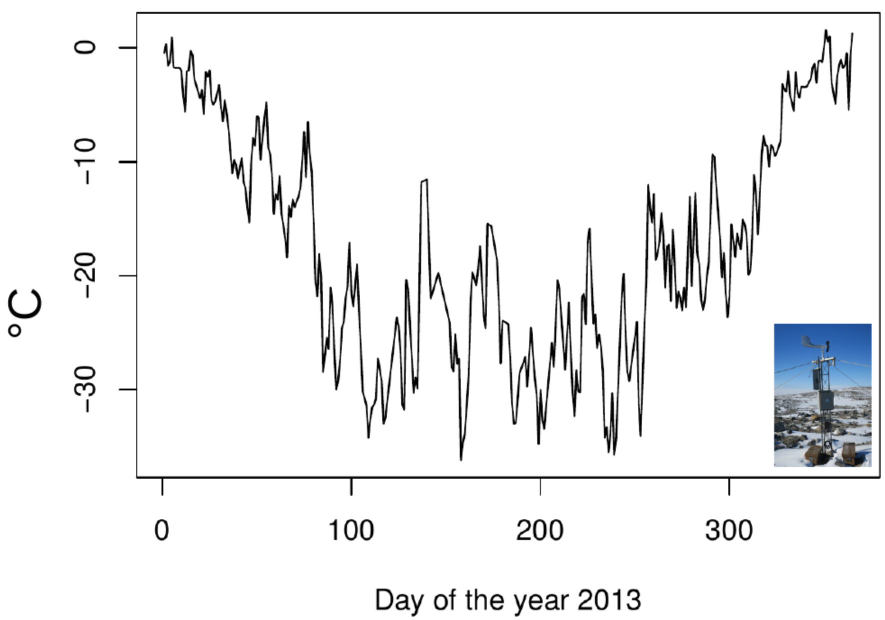
```
model <- train(predictors,
               response,
               method="rf",
               trControl=trainControl(method="cv"))
```

```
> model
Random Forest

30666 samples
   10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 27598, 27602, 27599, 27599, 27600, 27601, ...
Resampling results:

  RMSE      Rsquared
  5.554594  0.8986016

Tuning parameter 'mtry' was held constant at a value of 2
```

Problem with spatial (and temporal) data

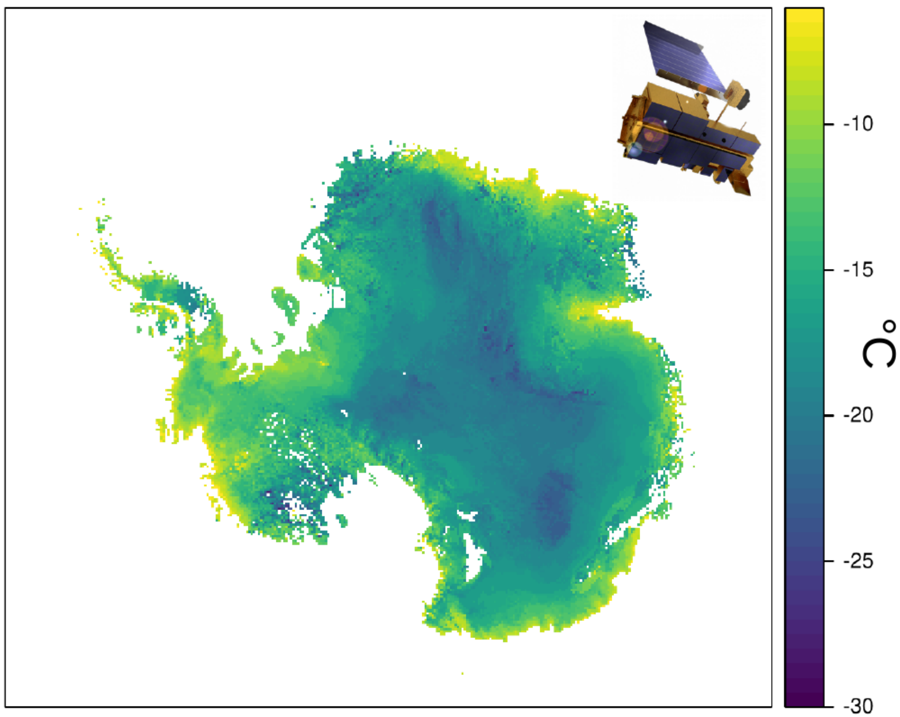
Temporal autocorrelation

Air temperature of "Marble Point"

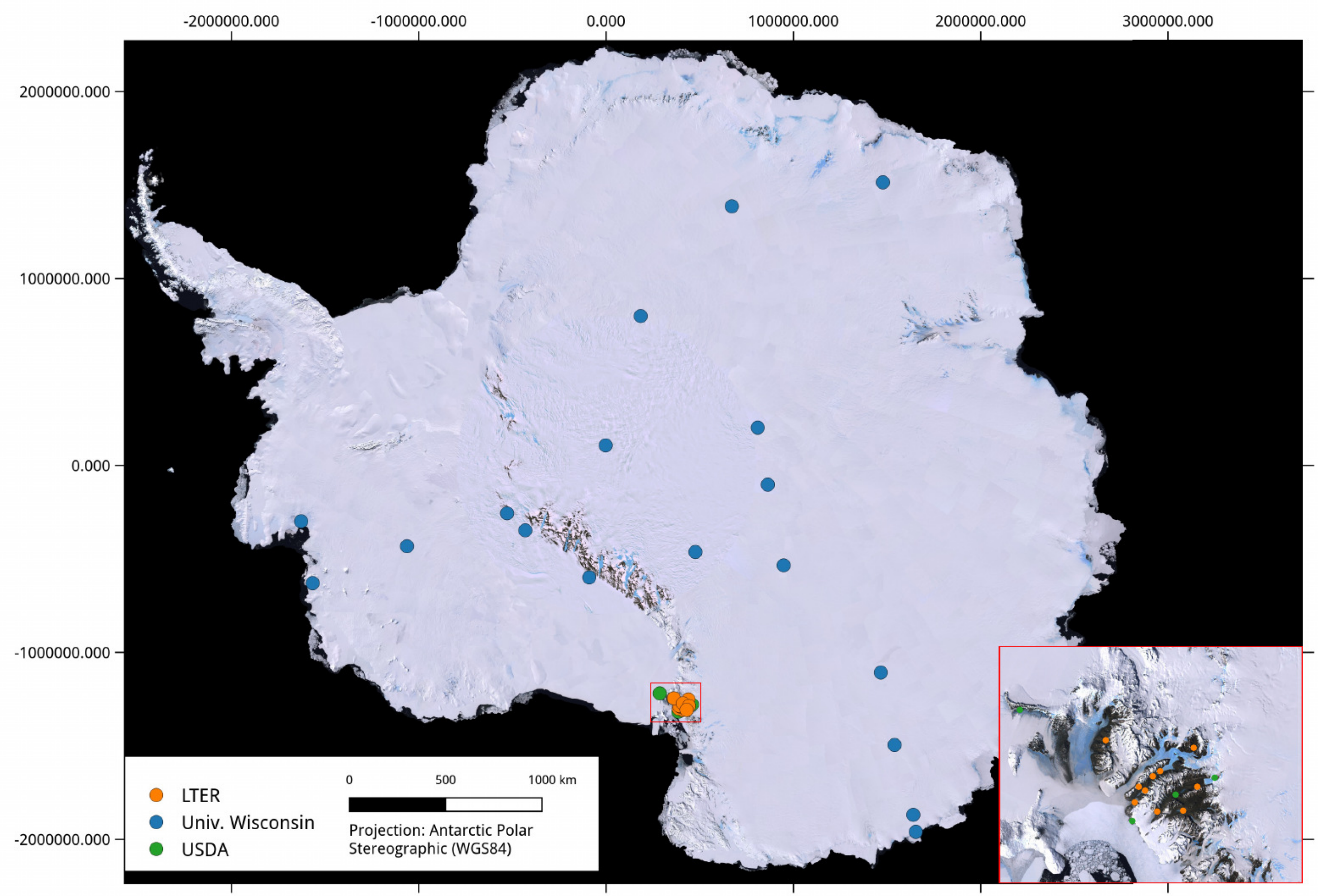


Spatial autocorrelation

MODIS LST from 2013/01/10

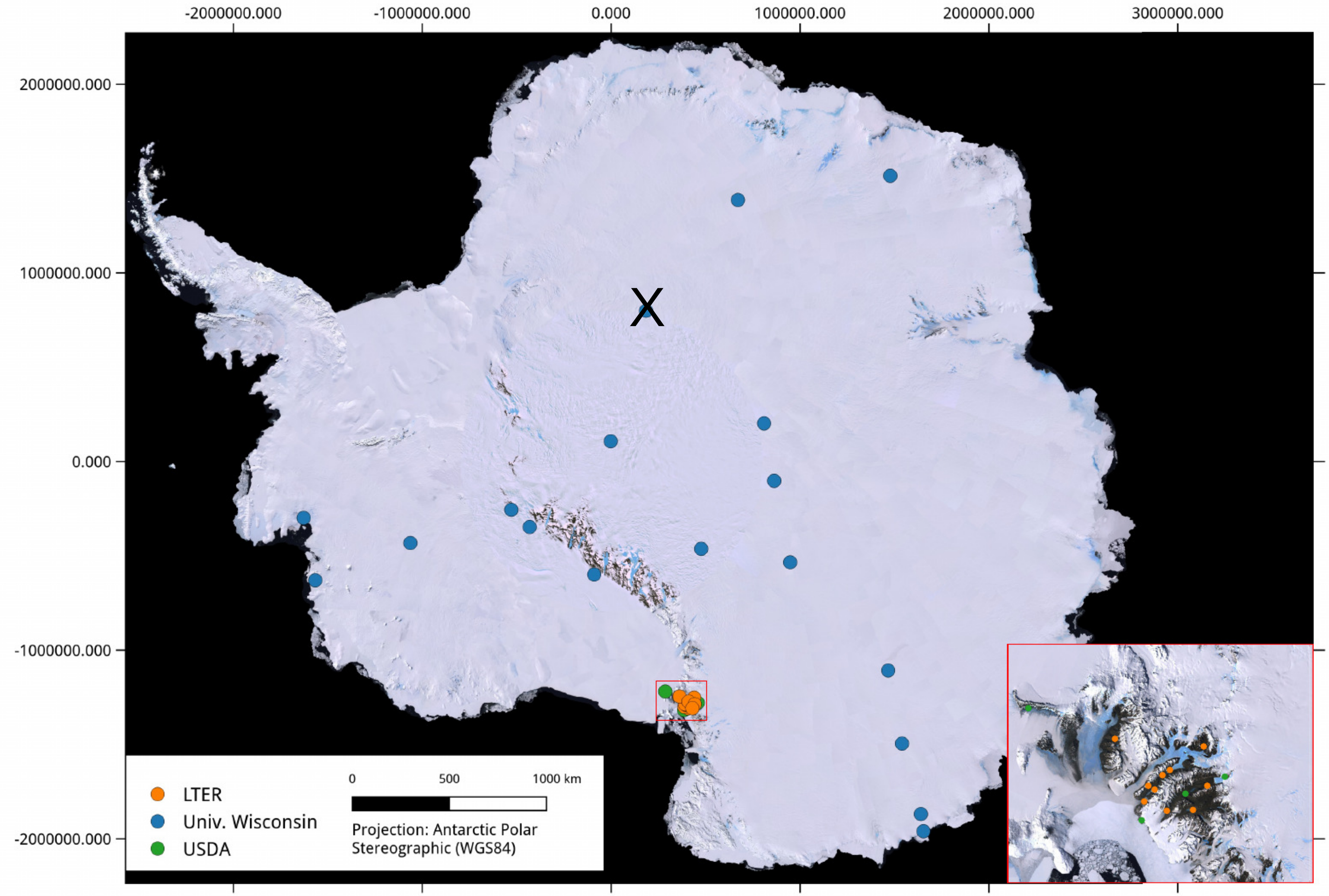


Target-oriented Cross-validation



Meyer et al. (2016)

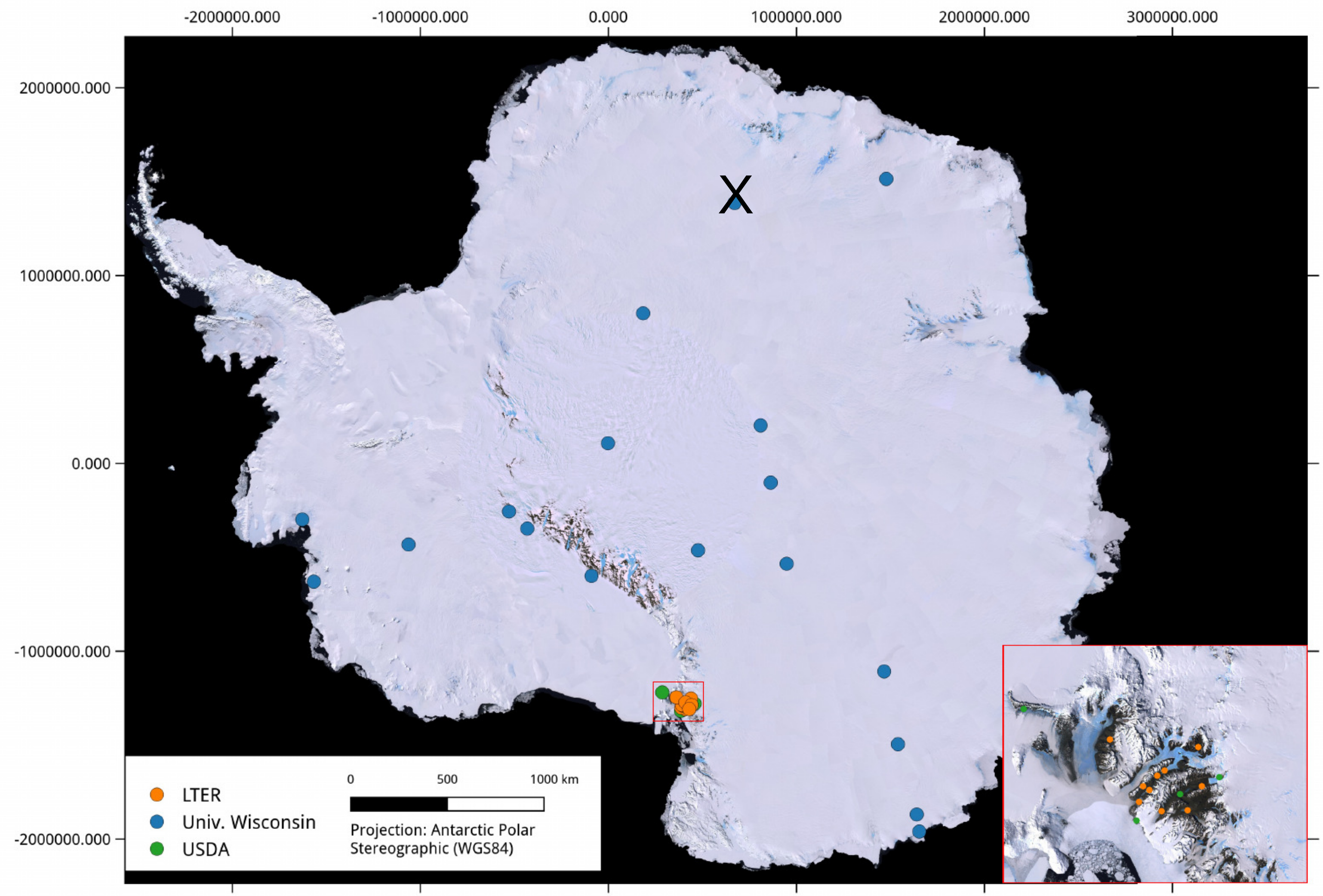
Target-oriented Cross-validation



Philipps Universität Marburg

Meyer et al. (2016)

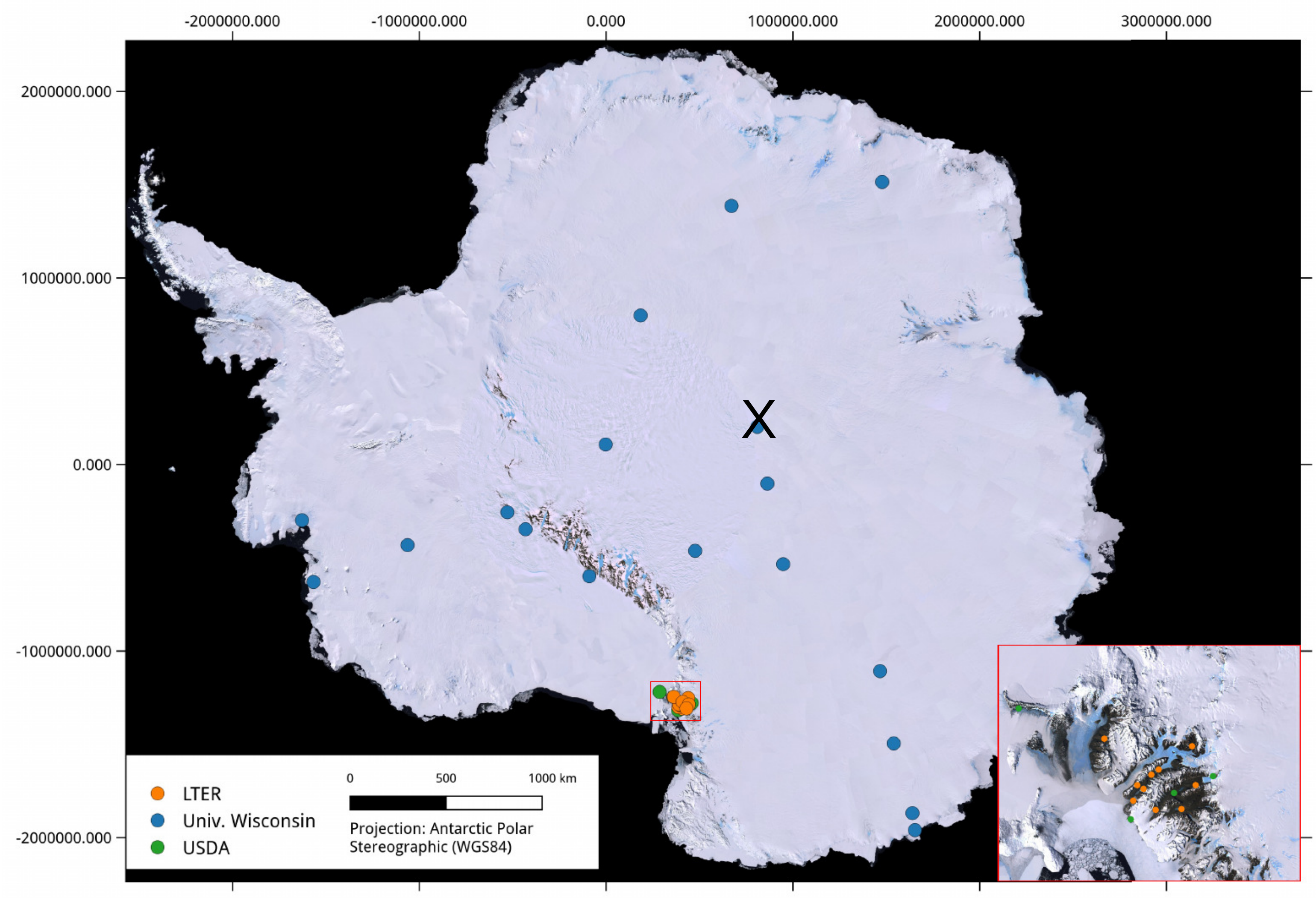
Target-oriented Cross-validation



Philipps Universität Marburg

Meyer et al. (2016)

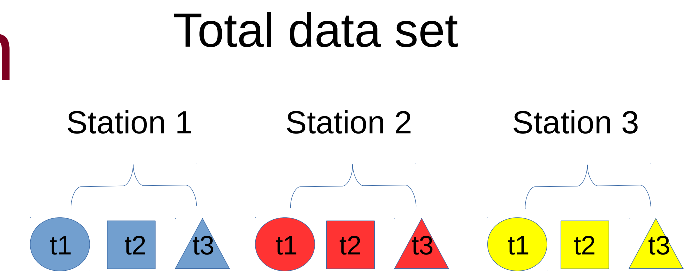
Target-oriented Cross-validation

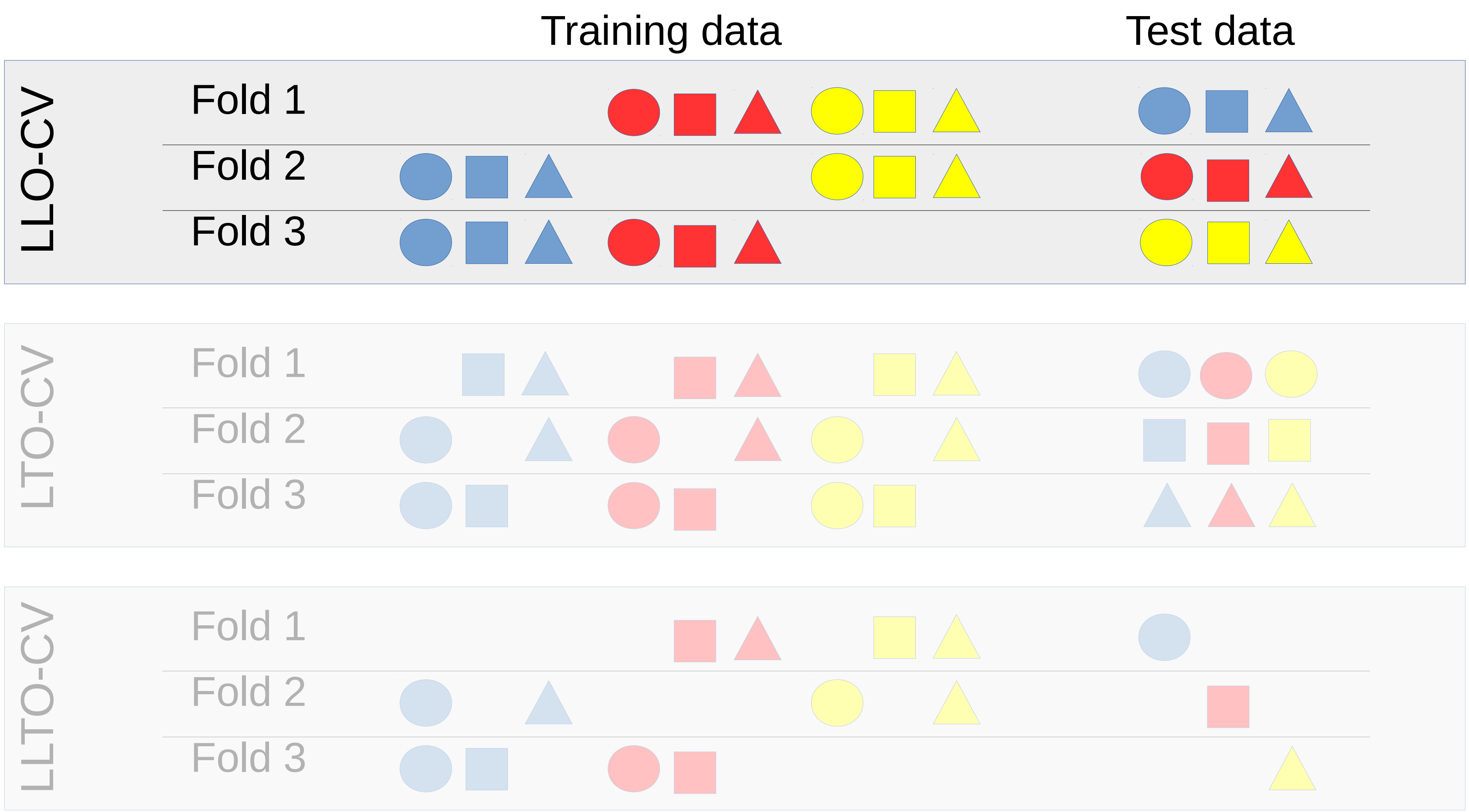


Philipps Universität Marburg

Meyer et al. (2016)

Target-oriented Cross-validation

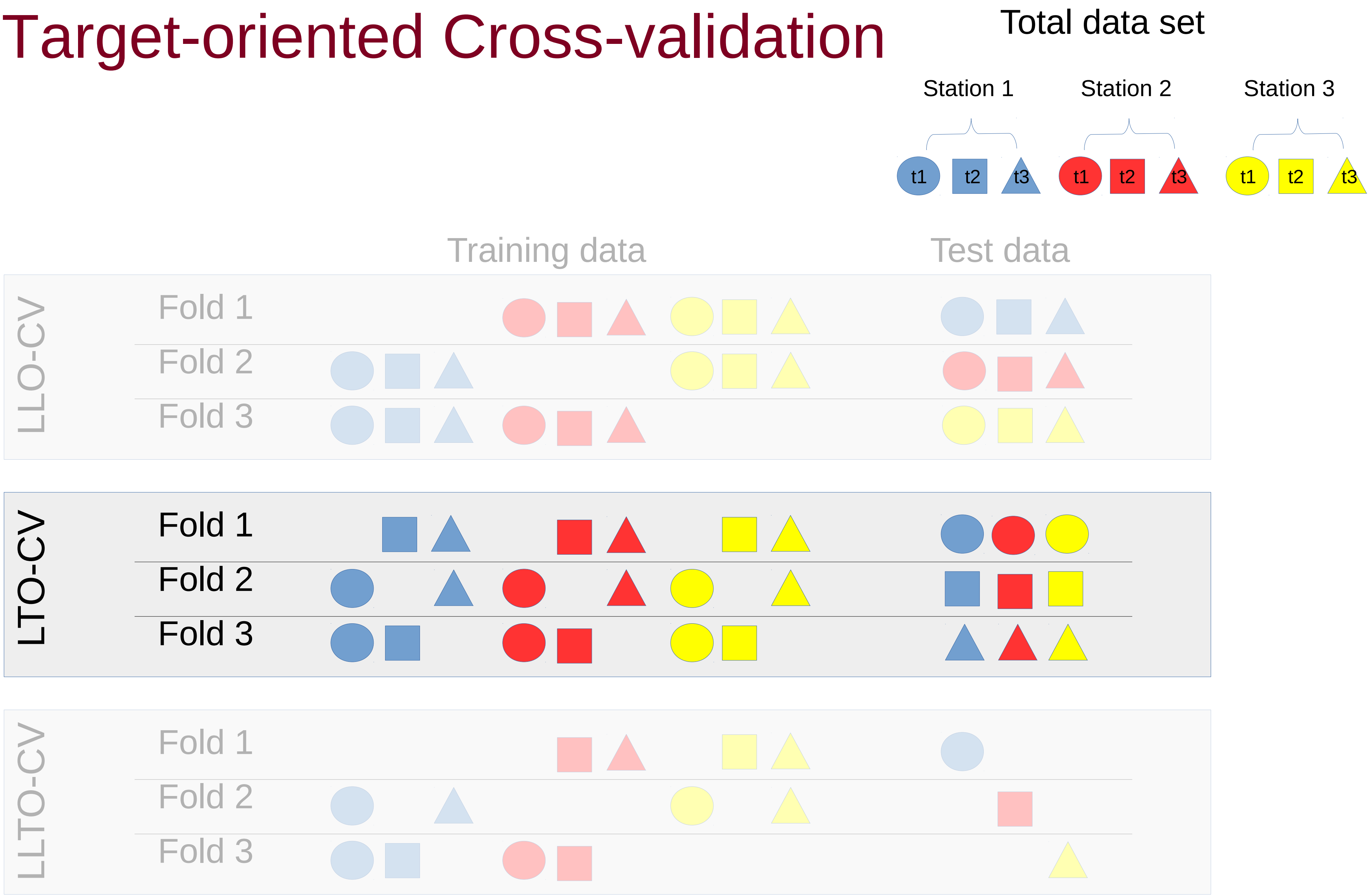




Philipps Universität Marburg

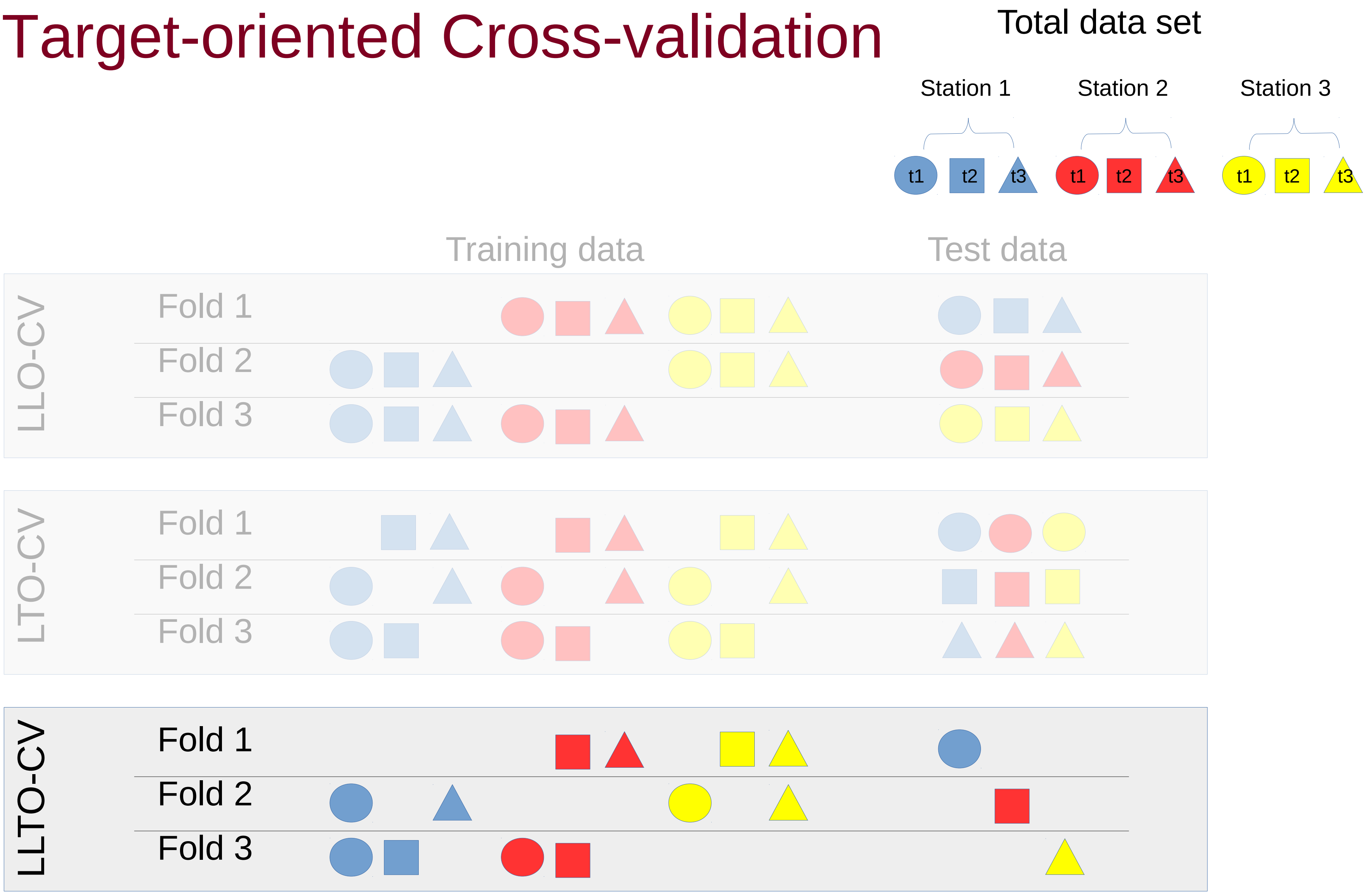
Meyer et al. (2018)

Target-oriented Cross-validation



Meyer et al. (2018)

Target-oriented Cross-validation



Meyer et al. (2018)

Target-oriented Cross-validation

How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- train(predictors,
               response,
               method="rf",
               trControl=trainControl(method="cv",
                                      index = indices$index))
```

Target-oriented Cross-validation

How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- train(predictors,
               response,
               method="rf",
               trControl=trainControl(method="cv",
                                      index = indices$index))
```

```
> model
Random Forest

30666 samples
   10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 27610, 28040, 26884, 27501, 28038,
Resampling results:

  RMSE      Rsquared
  14.52991  0.513388

Tuning parameter 'mtry' was held constant at a value of 2
```

Target-oriented Cross-validation

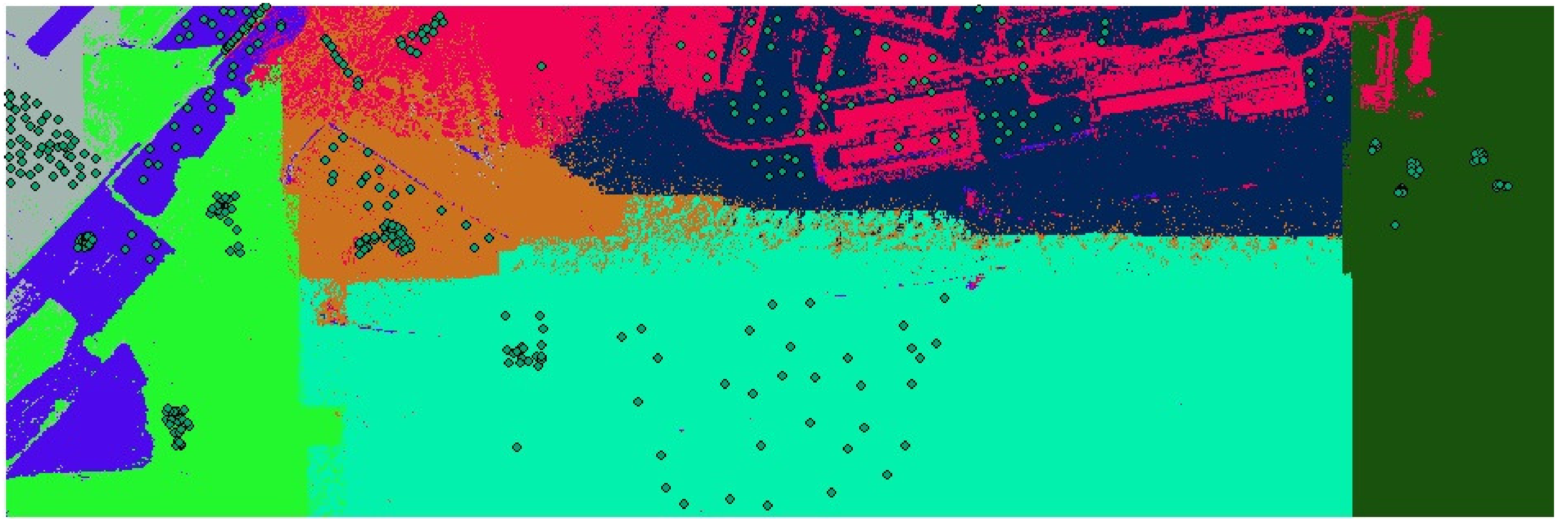
How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- train(predictors,
               response,
               method="rf",
               trControl=trainControl(method="cv",
                                      index = indices$index))
```

```
> model
Random Forest

30666 samples
   10 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 27610, 28040, 26884, 27501, 28038,
Resampling results:

  RMSE      Rsquared
  14.52991  0.513388

Tuning parameter 'mtry' was held constant at a value of 2
```

```
> summary(lm(model$pred$pred~model$pred$obs))

Call:
lm(formula = model$pred$pred ~ model$pred$obs)

Residuals:
    Min      1Q  Median      3Q     Max
-38.945  -6.416   1.898   8.771  29.176

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -16.927454   0.127665 -132.59   <2e-16 ***
model$pred$obs   0.389932   0.003918   99.52   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.84 on 30664 degrees of freedom
Multiple R-squared:  0.2441,    Adjusted R-squared:  0.2441
F-statistic:  9905 on 1 and 30664 DF,  p-value: < 2.2e-16
```

Overfitting due to misinterpretations of variables

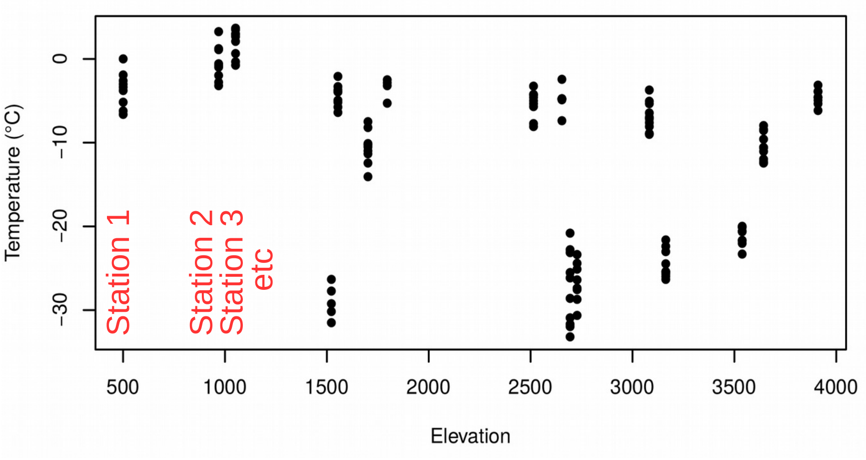


https://gis.stackexchange.com/questions/111932/classified-images-of-randomforest-classification-look-clustered

Overfitting due to misinterpretations of variables

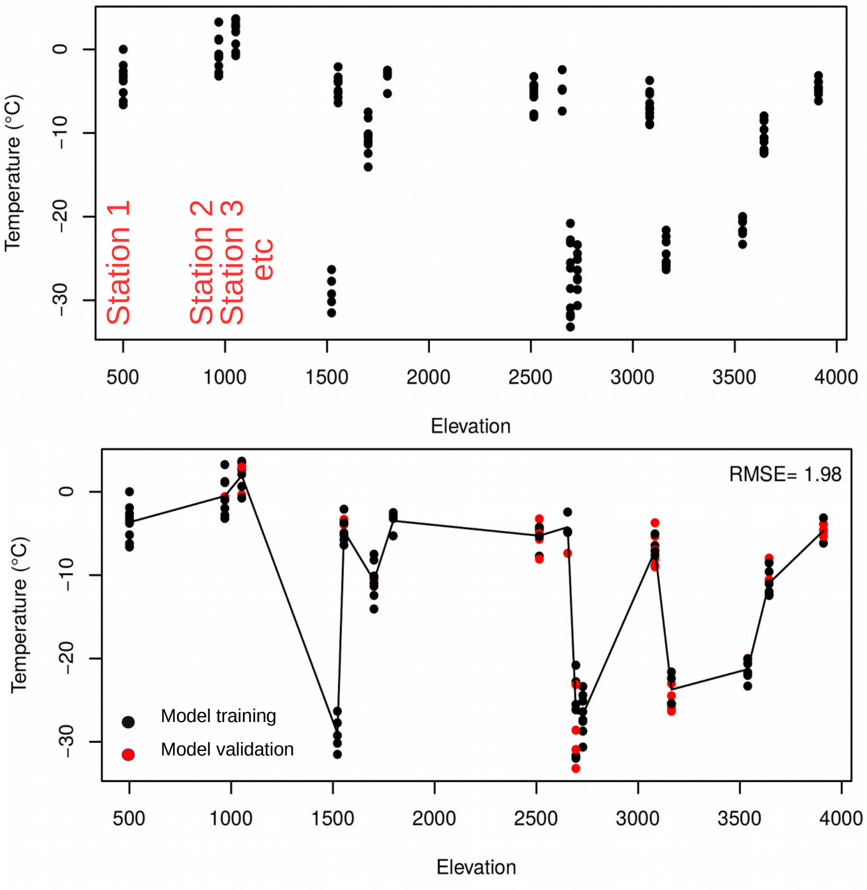
Station	Date	LST	**Elevation**	Aspect	...	Measured Tair
A	2017/01/01	-5	1000	S		-2
B	2017/01/01	0	200	S		-2
C	2017/01/01	-10	3000	E		-5
A	2017/07/01	-40	1000	S		-45
B	2017/07/01	-30	200	S		-30
C	2017/07/01	-60	3000	E		-70
A	2017/10/01	-20	1000	S		-22
B	2017/10/01	-10	200	S		-9
C	2017/10/01	-25	3000	E		-30

Overfitting due to misinterpretations of variables



Unique spatial variable for each location (e.g. elevation)

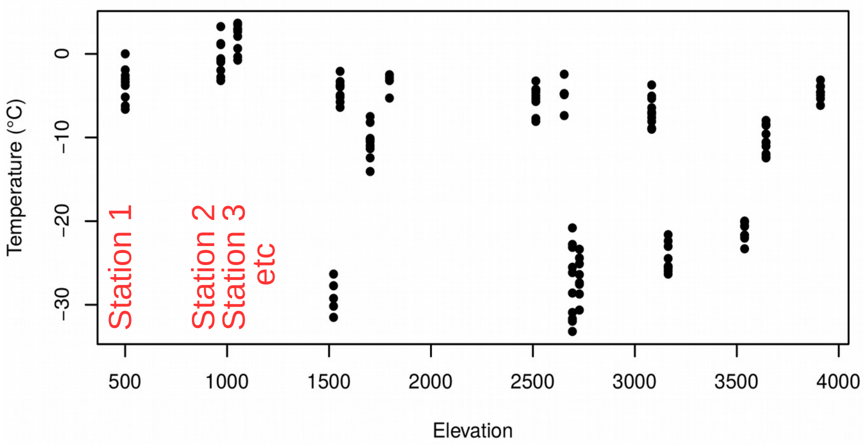
Overfitting due to misinterpretations of variables



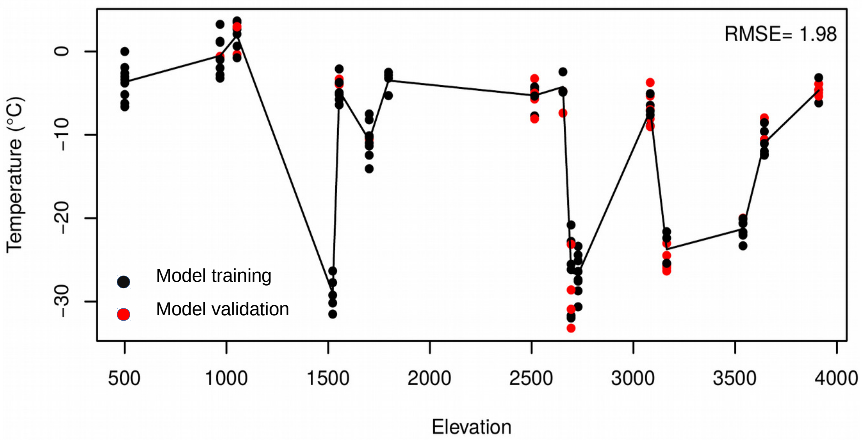
Unique spatial variable for each location (e.g. elevation)

Internal result: elevation is important for the model

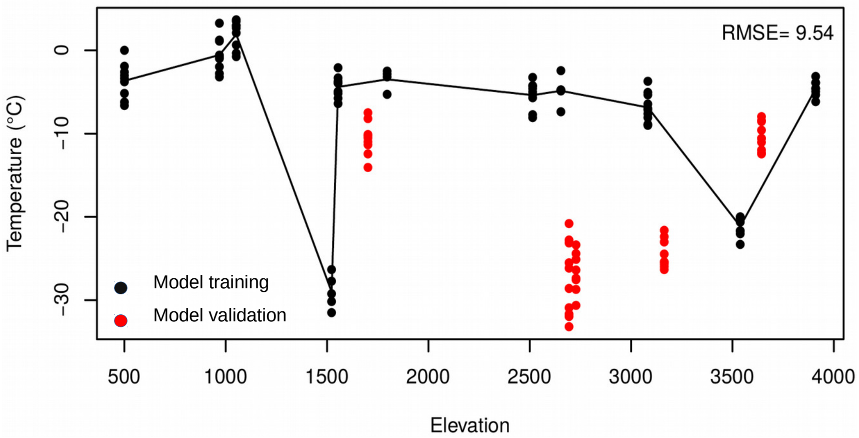
Overfitting due to misinterpretations of variables



Unique spatial variable for each location (e.g. elevation)



Internal result: elevation is important for the model



...but importance originates from ability of the algorithm to access the individual time series and not from spatial meaning

“Target-oriented” variable selection

```
for each resampling iteration do
    Partition the data into training and test data
    Tune and train models using all possible 2-variable combinations
    Predict on test data and calculate model performance
end
Keep the best performing 2-variable model (model_best)
for each additional number of variables i, i=3...N do
    for each remaining variable V_R do
        for each resampling iteration do
            Partition the data into training and test data
            Tune and train models using the variables of model_best and V_R
            Predict on test data and calculate model performance
        end
    end
    if mean(error of model_i) > mean(error of model_best) then
        break
    end
    Keep the best performing i-variable model (model_best)
end
```

LLO cross-validation!

Which 2 variables lead to the best model?

LLO cross-validation!

Which further variables improve the model?

“Target-oriented” variable selection in R

How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- ffs(predictors,
             response,
             method="rf",
             trControl=trainControl(method="cv",
                                    index = indices$index))
```

“Target-oriented” variable selection in R

How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- ffs(predictors,
             response,
             method="rf",
             trControl=trainControl(method="cv",
                                    index = indices$index))
```

```
> model
Random Forest

30666 samples
    5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 27610, 28040, 26884, 27501, 28038,
Resampling results:

  RMSE      Rsquared
  11.60326  0.596083

Tuning parameter 'mtry' was held constant at a value of 2
```

“Target-oriented” variable selection in R

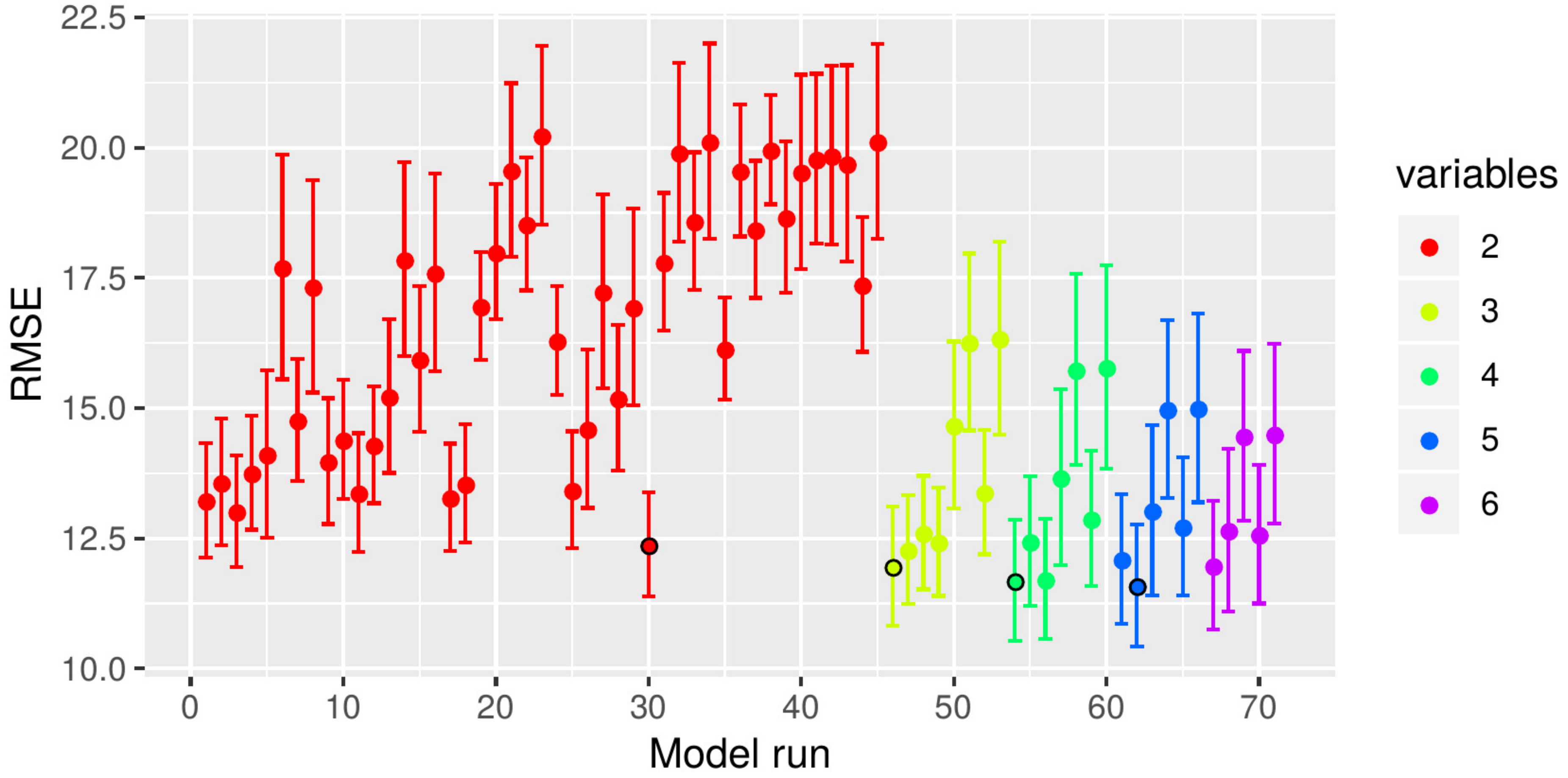
How to do it in R

```
library(CAST)
indices <- CreateSpacetimeFolds(trainingData,
                                spacevar="Station")
model <- ffs(predictors,
             response,
             method="rf",
             trControl=trainControl(method="cv",
                                    index = indices$index))
```

```
> model
Random Forest

30666 samples
    5 predictor

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 27610, 28040, 26884, 27501, 28038,
Resampling results:

  RMSE      Rsquared
  11.60326  0.596083

Tuning parameter 'mtry' was held constant at a value of 2
```

```
> summary(lm(model$pred$pred~model$pred$obs))

Call:
lm(formula = model$pred$pred ~ model$pred$obs)

Residuals:
     Min       1Q   Median       3Q      Max
-31.0891  -5.1795   0.6637   6.7039  30.3651

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -13.433596   0.101226  -132.7   <2e-16 ***
model$pred$obs   0.516511   0.003107   166.3   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.391 on 30664 degrees of freedom
Multiple R-squared:  0.4741,	Adjusted R-squared:  0.4741
F-statistic: 2.764e+04 on 1 and 30664 DF,  p-value: < 2.2e-16
```

“Target-oriented” variable selection in R



Selected variables: “LST”, "month", "ice", "season", "sensor"

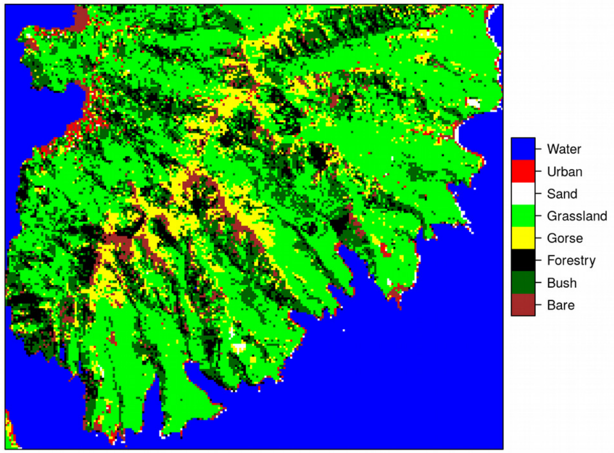
Take home messages

- ML has great potential for spatial and spatio-temporal predictions
- Caret allows for easy model training
- But spatial data need adapted ML frameworks
- Target-oriented error assessment is important: LLO as shown here or block CV (Roberts 2015), spatial CV (Brenning 2012)
- Risk of overfitting by misinterpretation of predictor variables
- Avoid overfitting by careful variable selection (e.g. automatically via CAST)

Outlook computer practice

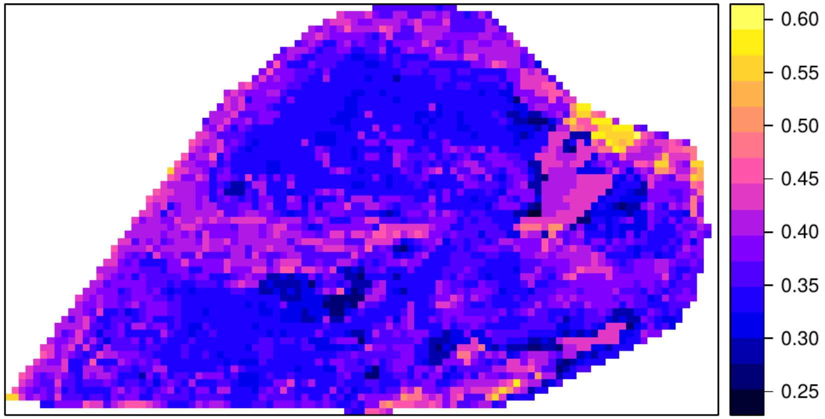
Case Study I: Land cover classification

- Task: Identify the invasive gorse on the Banks Peninsula in New Zealand based on Sentinel satellite data
- Technical Challenge: “Simple” ML model training for spatial predictions



Case Study II: Spatio-temporal predictions

- Task: Model soil moisture in space and time for the cookfarm
- Technical Focus: Sensitivity of ML to different CV strategies and variable selection



References

- Brenning, A. 2012. Spatial cross-validation and bootstrap for the assessment of prediction rules in remote sensing: the R package 'sperrorest'. IEEE International Symposium on Geoscience and Remote Sensing IGARSS.
- Kuhn, M. and K. Johnson, 2013a: Applied Predictive Modeling. 1st ed., Springer, New York.
- Kuhn, M. (2018): caret: Classification and Regression Training. R package version 6.0-80. https://github.com/topepo/caret/.
- Meyer, H., Katurji, M., Appelhans, T., Müller, M.U., Nauss, T., Roudier, P., Zawar-Reza, P. (2016): Mapping daily air temperature for Antarctica based on MODIS LST. Remote Sensing, 8(9), 732.
- Meyer, H., Reudenbach, C., Hengl, T., Katurji, M., Nauss, T. (2018): Improving performance of spatio-temporal machine learning models using forward feature selection and target-oriented validation. Environmental Modelling & Software, 101, 1-9.
- Meyer, H. (2018). CAST: 'caret' Applications for Spatial-Temporal Models. R package version 0.2.1. https://CRAN.R-project.org/package=CAST.
- Roberts, D. R., V. Bahn, S. Ciuti, M. S. Boyce, J. Elith, G. Guillera-Arroita, S. Hauenstein, J. J. Lahoz-Monfort, B. Schröder, W. Thuiller, D. I. Warton, B. A. Wintle, F. Hartig, and C. F. Dormann, 2017: Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography. 40(8): 913-929.