Division of
Statistics + Scientific Computation

THE UNIVERSITY OF TEXAS AT AUSTIN

Advanced Regression
Summer Statistics Institute

Day 4: Time Series, Logistic Regression and More...

Time Series Data and Dependence

Time-series data are simply a collection of observations gathered over time. For example, suppose $y_1 \dots y_T$ are

- Annual GDP.
- Quarterly production levels
- Weekly sales.
- Daily temperature.
- 5 minute Stock returns.

In each case, we might expect what happens at time t to be correlated with what happens at time $t-1$.

Time Series Data and Dependence

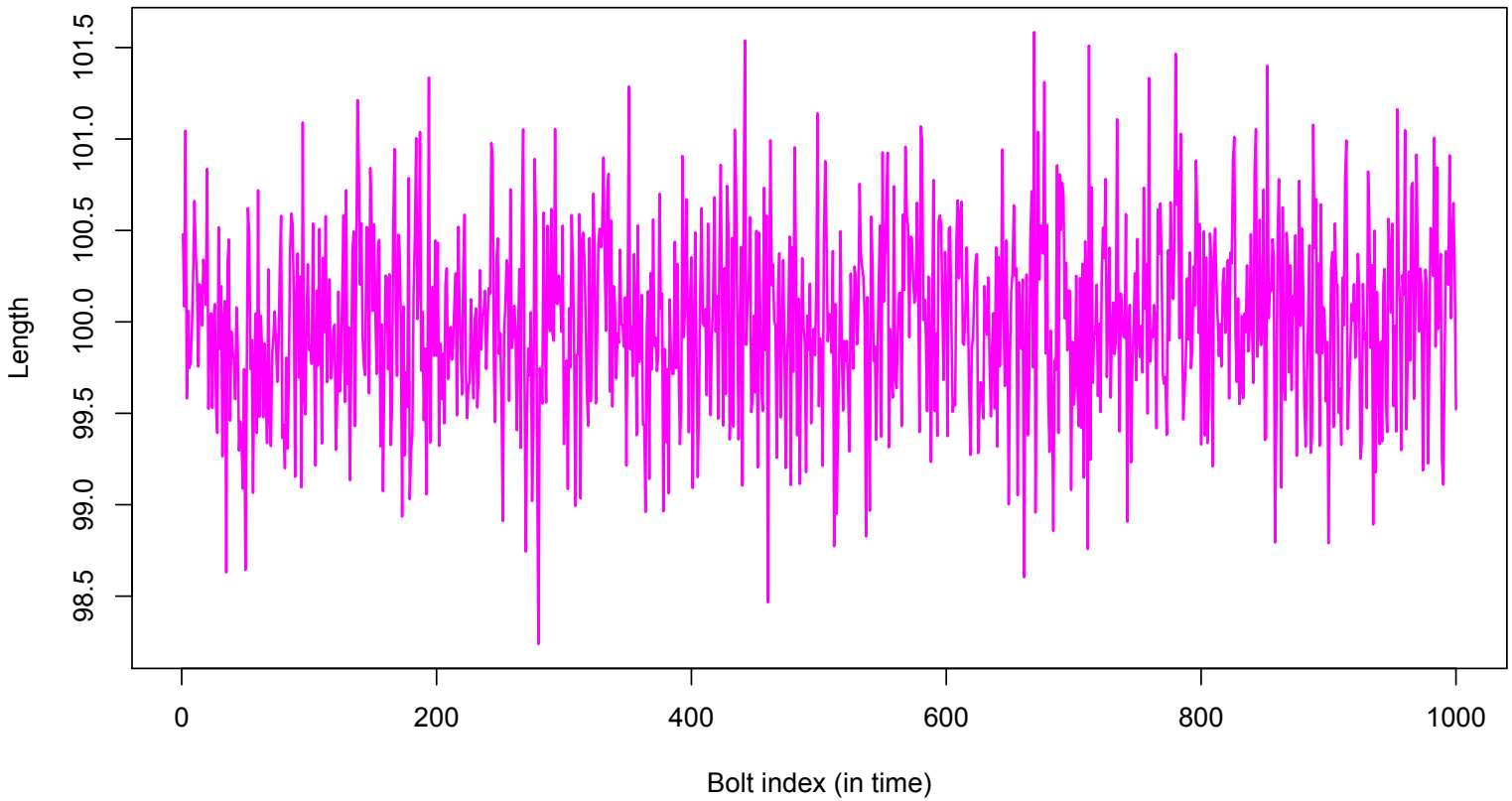
Suppose we measure temperatures daily for several years.

Which would work better as an estimate for today's temp:

- The average of the temperatures from the previous year?
- The temperature on the previous day?

Example: Length of a bolt...

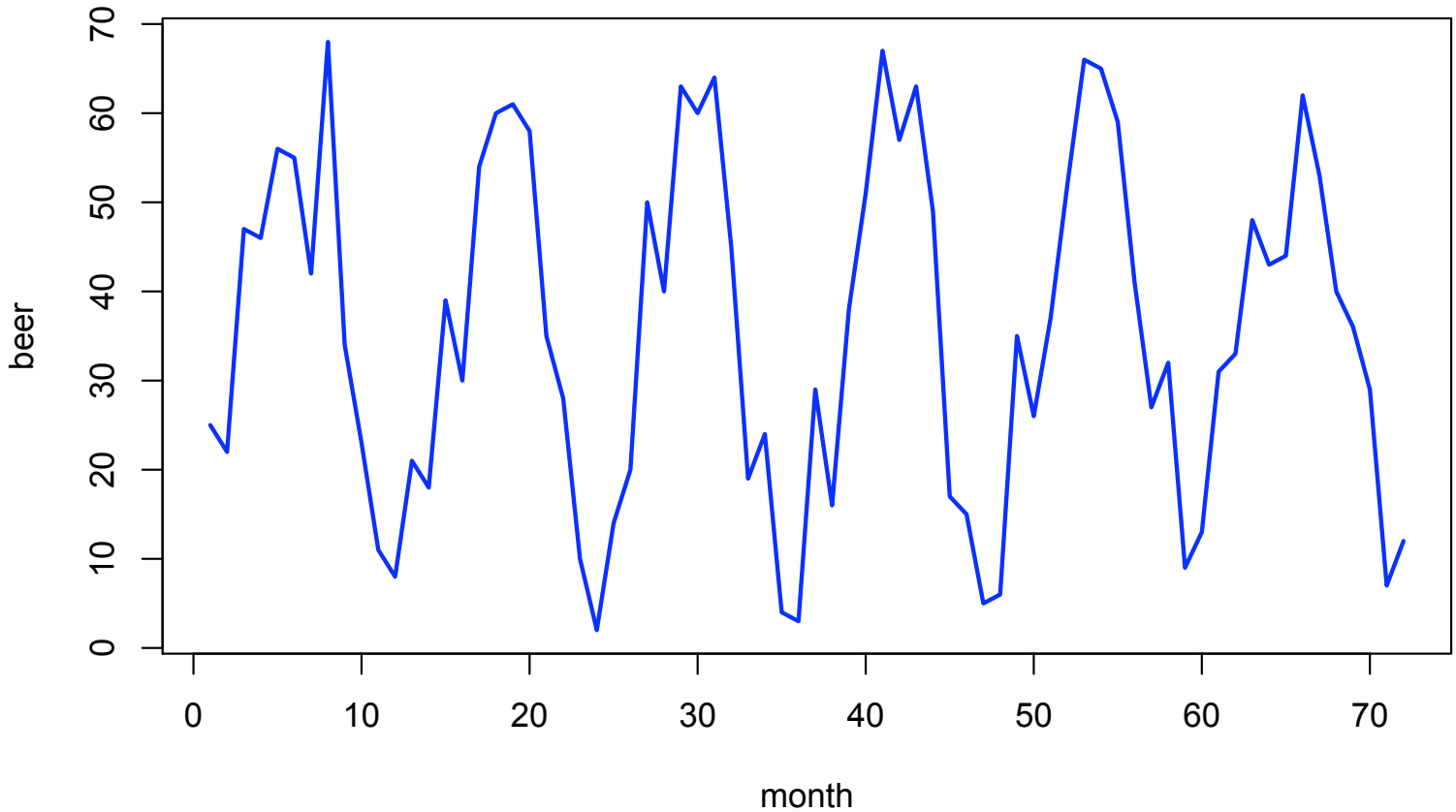
Suppose you have to check the performance of a machine making bolts... in order to do so you want to predict the length of the next bolt produced...



What is your best guess for the next part?

Example: Beer Production

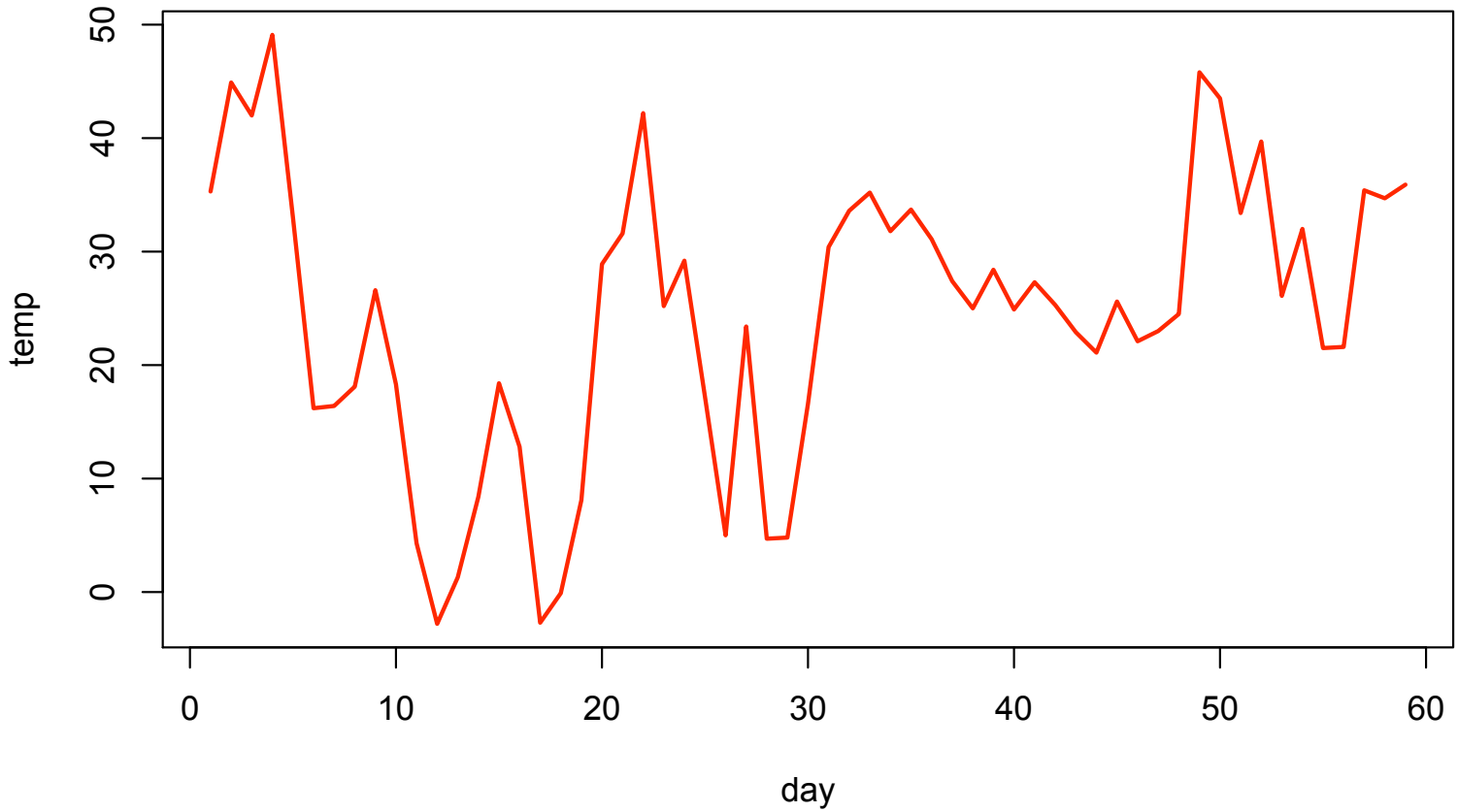
Now, say you want to predict the monthly U.S. beer production (in millions of barrels).



What about now, what is your best guess for the production in the next month?

Examples: Temperatures

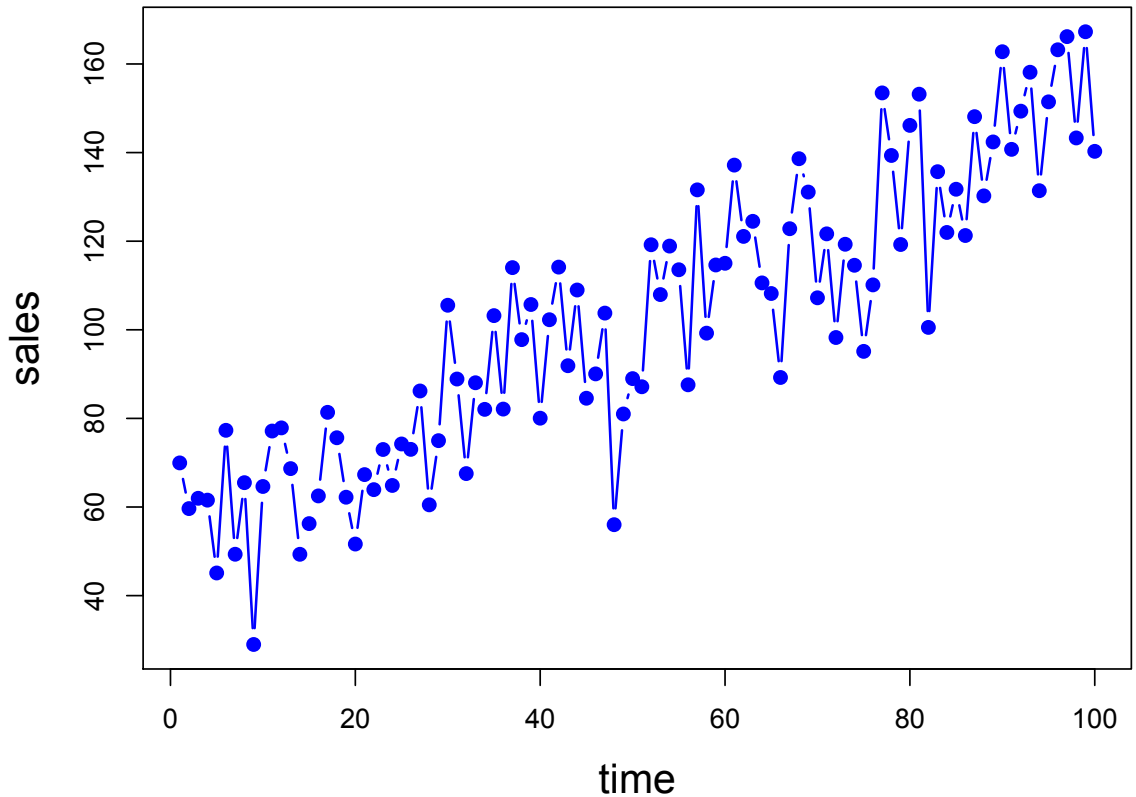
Now you need to predict tomorrow's temperature at O'Hare from (Jan-Feb).



Is this one harder? Our goal in this section is to use regression models to help answer these questions...

Fitting a Trend

Here's a time series plot of monthly sales of a company...



What would be a reasonable prediction for Sales 5 months from now?

Fitting a Trend

The sales numbers are "trending" upwards... What model could capture this trend?

$$S_t = \beta_0 + \beta_1 t + \epsilon_t \qquad \epsilon_t \sim N(0, \sigma^2)$$

This is a regression of Sales (y variable) on "time" (x variable).
This allows for shifts in the mean of Sales as a function of time.

Fitting a Trend

The data for this regression looks like:

months(t)	Sales
1	69.95
2	59.64
3	61.96
4	61.55
5	45.10
6	77.31
7	49.33
8	65.49
...	...
100	140.27

Fitting a Trend

$$S_t = \beta_0 + \beta_1 t + \epsilon_t \qquad \epsilon_t \sim N(0, \sigma^2)$$

Regression Statistics	
Multiple R	0.892
R Square	0.796
Adjusted R Square	0.794
Standard Error	14.737
Observations	100.000

ANOVA

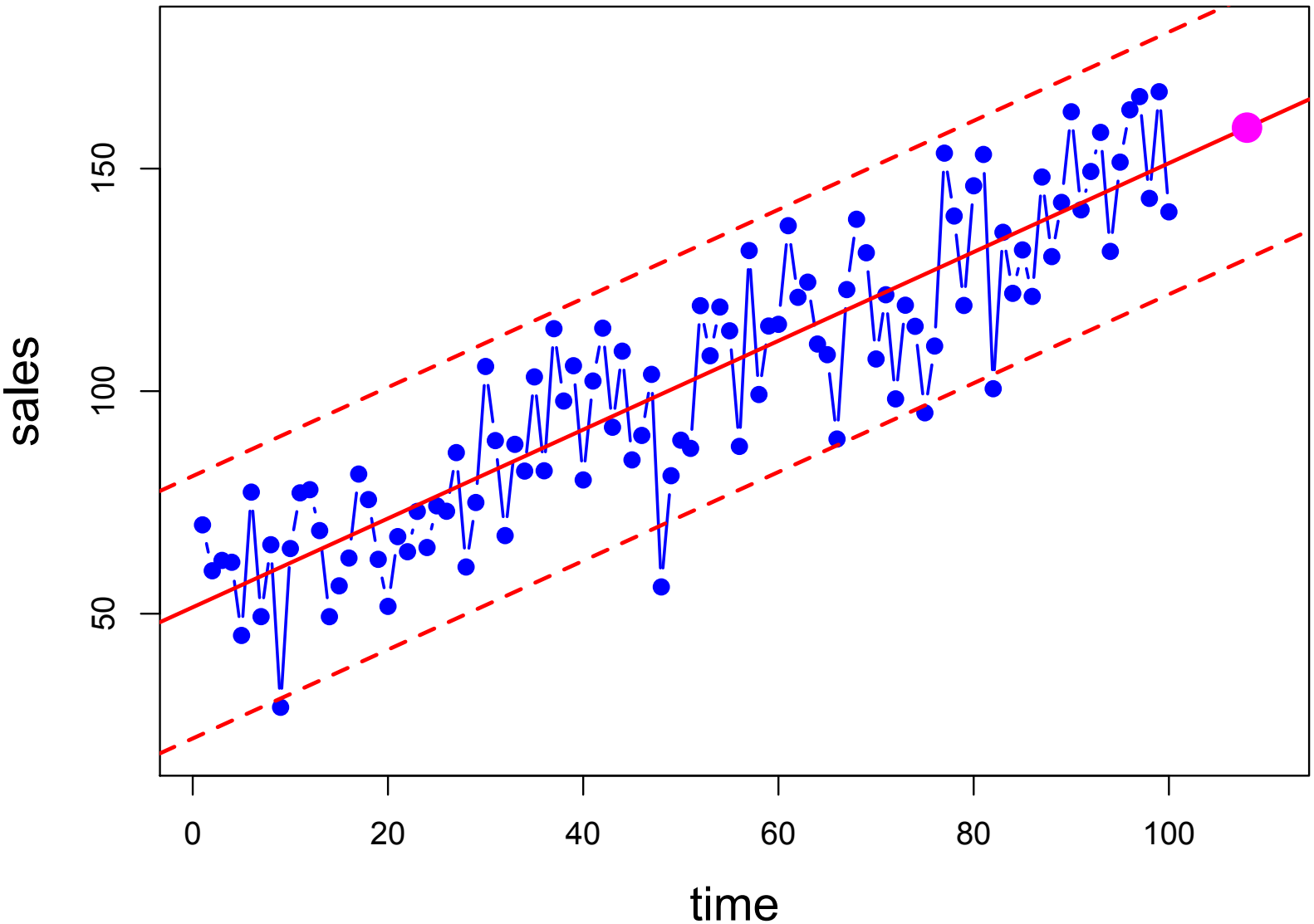
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1.000	82951.076	82951.076	381.944	0.000
Residual	98.000	21283.736	217.181		
Total	99.000	104234.812			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	51.442	2.970	17.323	0.000	45.549	57.335
t	0.998	0.051	19.543	0.000	0.896	1.099

$$S_t = 51.44 + 0.998t \pm 2 * 14.73$$

Fitting a Trend

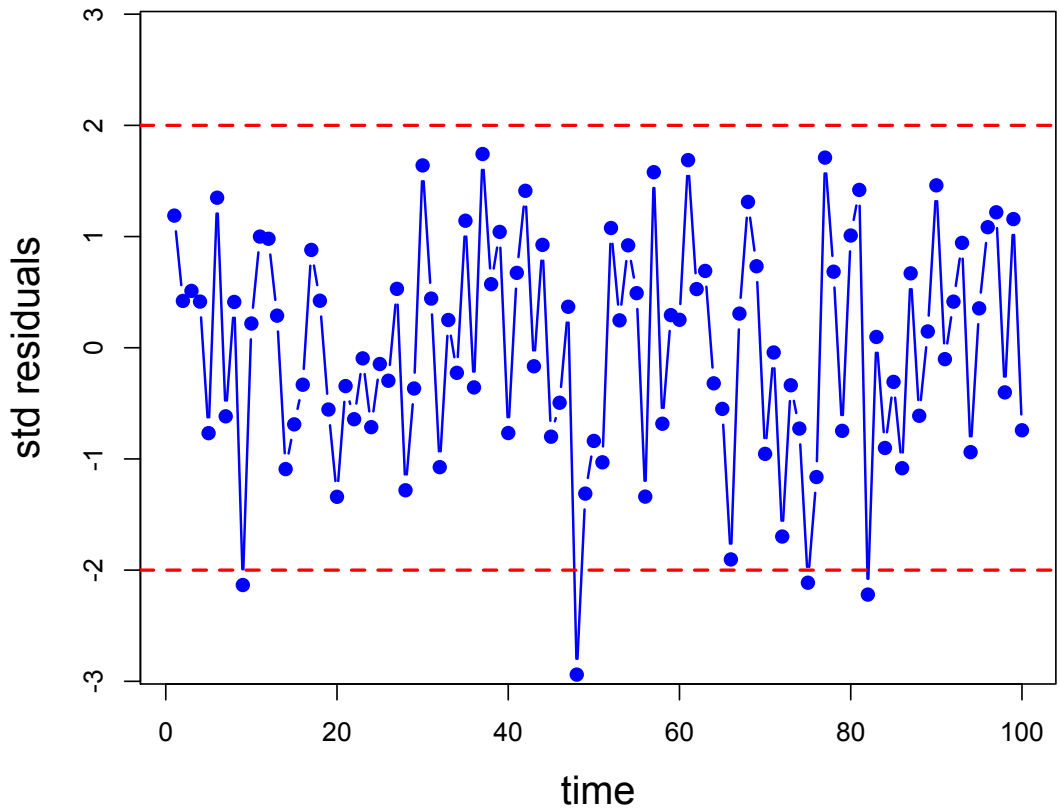
Plug-in prediction...



Residuals

How should our residuals look? If our model is correct, the trend

should have captured the time series structure is sales and what is left, should not be associated with time... i.e., it should be iid normal.



Great!

Time Series Regression... Hotel Occupancy Case

In a recent legal case, a Chicago downtown hotel claimed that it had suffered a loss of business due to what was considered an illegal action by a group of hotels that decided to leave the plaintiff out of a hotel directory.

In order to estimate the loss business, the hotel had to predict what its level of business (in terms of occupancy rate) would have been in the absence of the alleged illegal action.

In order to do this, experts testifying on behalf of the hotel use data collected before the period in question and fit a relationship between the hotels occupancy rate and overall occupancy rate in the city of Chicago. This relationship would then be used to predict occupancy rate during the period in question.

Example: Hotel Occupancy Case

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.7111011
R Square	0.5056648
Adjusted R Squa	0.48801
Standard Error	7.5055176
Observations	30

$$Hotel_t = \beta_0 + \beta_1 Chicago + \epsilon_t$$

ANOVA

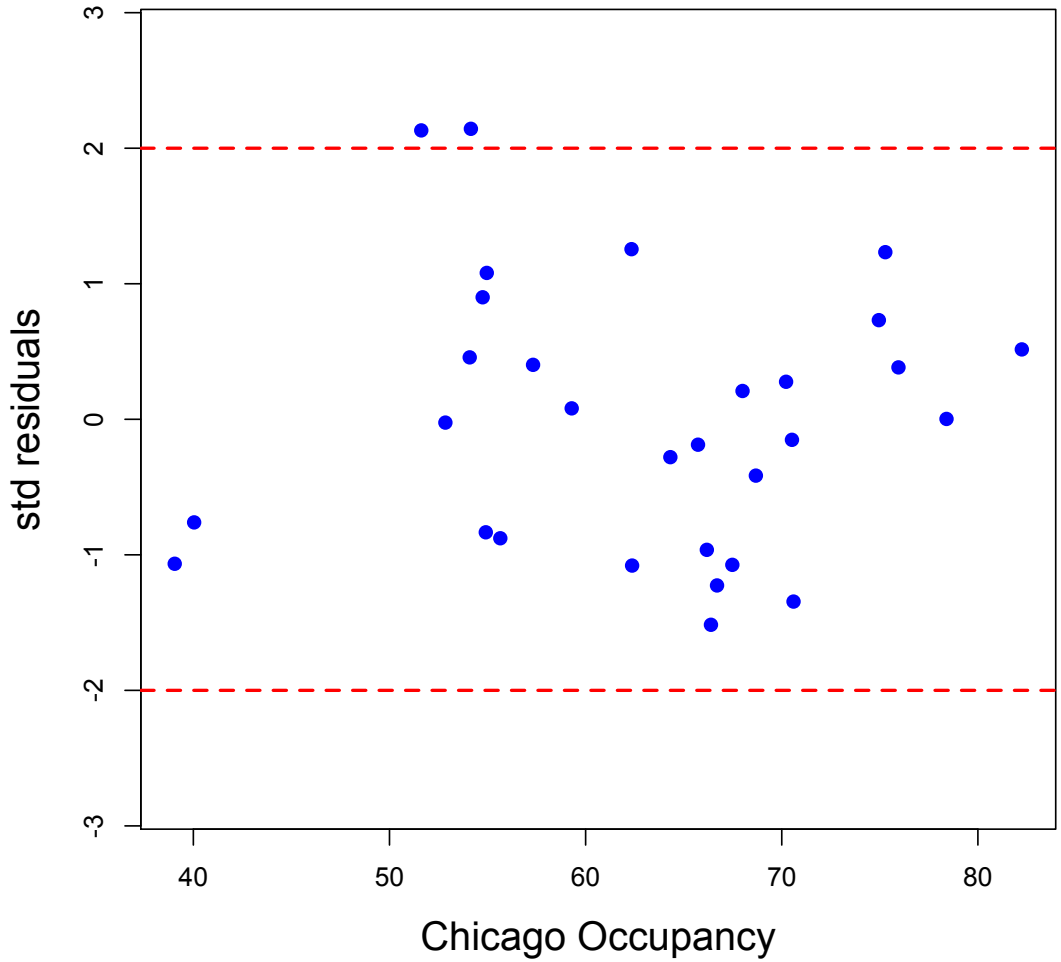
	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	1613.468442	1613.4684	28.64172598	1.06082E-05
Residual	28	1577.318225	56.332794		
Total	29	3190.786667			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	16.135666	8.518889357	1.8941044	0.068584205	-1.314487337	33.5858198
ChicagoInd	0.7161318	0.133811486	5.3517965	1.06082E-05	0.442031445	0.990232246

- In the month after the omission from the directory the Chicago occupancy rate was 66. The plaintiff claims that its occupancy rate should have been 16 + 0.71*66 = 62.
- It was actually 55!! The difference added up to a big loss!!

Example: Hotel Occupancy Case

A statistician was hired by the directory to access the regression methodology used to justify the claim. As we should know by now, the first thing he looked at was the residual plot...



Looks fine. However...

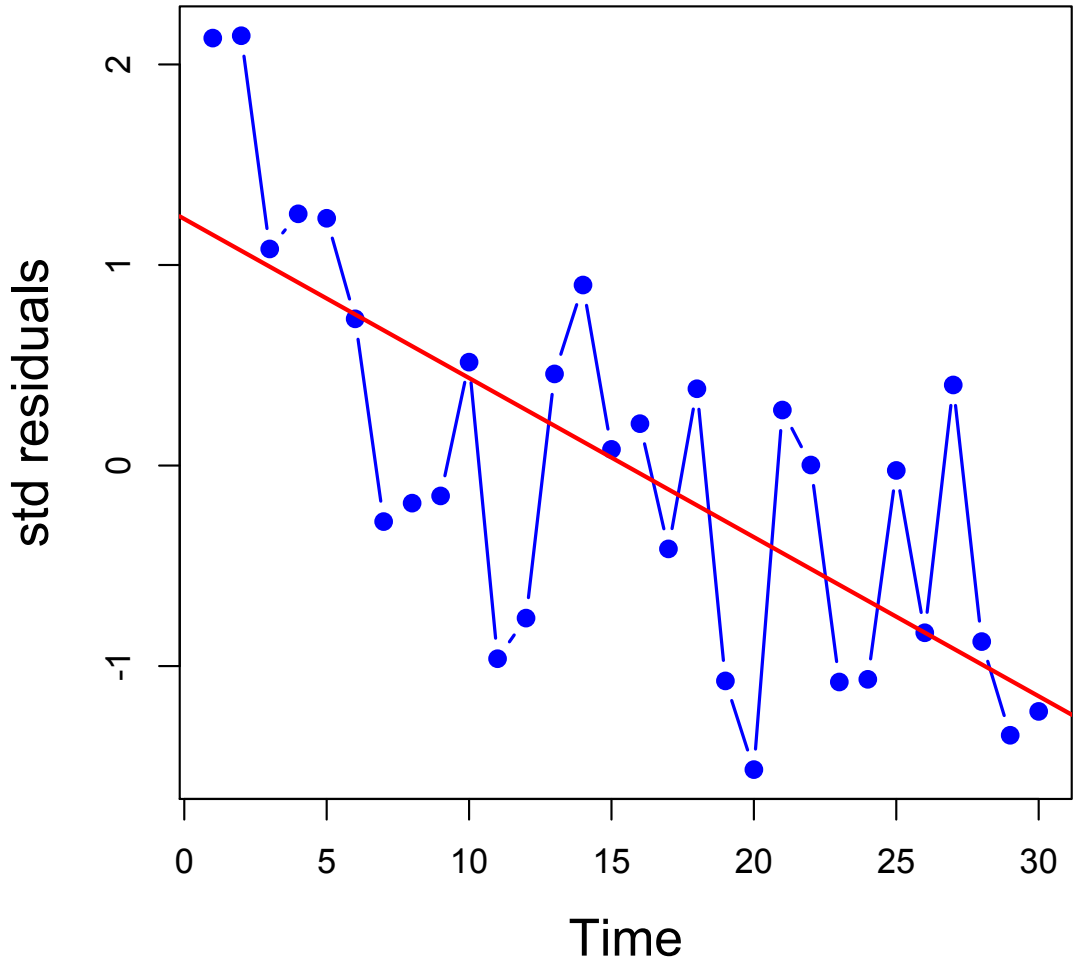
Example: Hotel Occupancy Case

... this is a *time series regression*, as we are regressing one time series on another.

In this case, we should also check whether or not the residuals show some temporal pattern.

If our model is correct the residuals should look iid normal over time.

Example: Hotel Occupancy Case



Does this look iid to you? Can you guess what does the red line represent?

Example: Hotel Occupancy Case

It looks like part of hotel occupancy (y) not explained by the Chicago downtown occupancy (x) is moving down over time. We can try to control for that by adding a trend factor to our model...

$$Hotel_t = \beta_0 + \beta_1 Chicago + \beta_2 t + \epsilon_t$$

SUMMARY OUTPUT

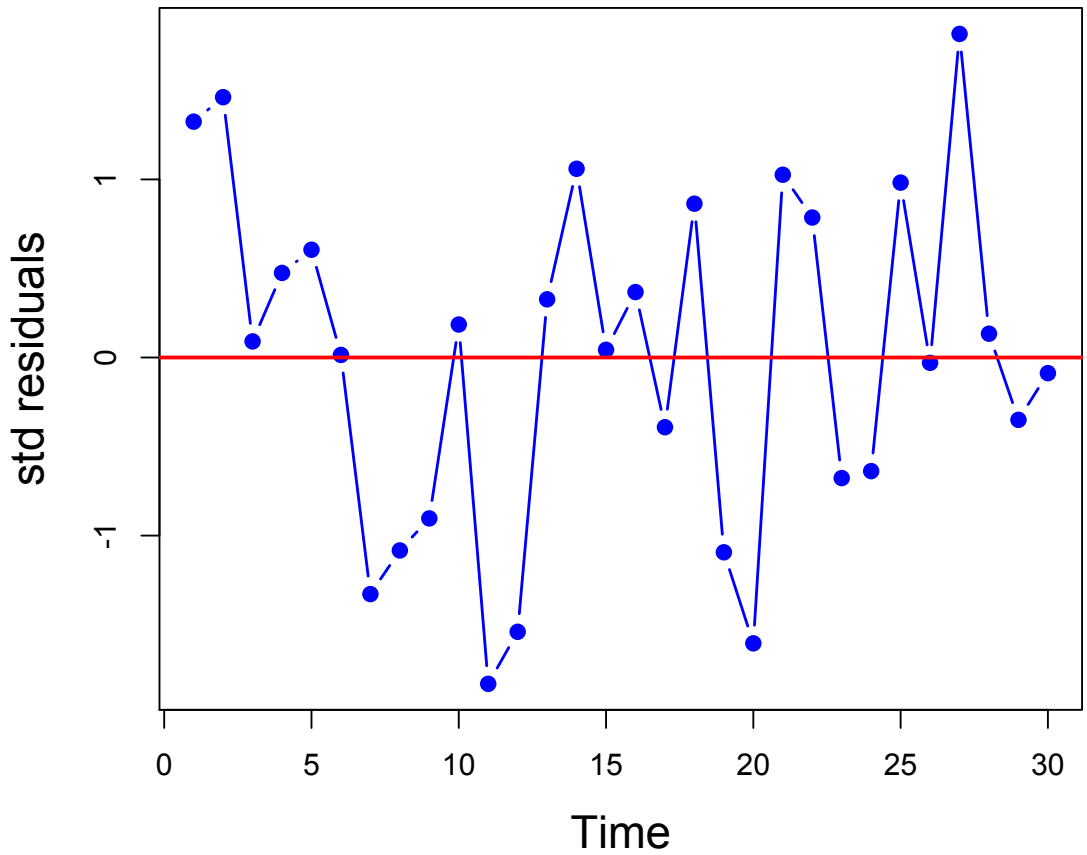
Regression Statistics	
Multiple R	0.869389917
R Square	0.755838827
Adjusted R Squ	0.737752815
Standard Error	5.37162026
Observations	30

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	2411.720453	1205.86	41.79134652	5.41544E-09
Residual	27	779.0662139	28.8543		
Total	29	3190.786667			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	26.69391108	6.418837165	4.158683	0.000290493	13.52354525	39.8642769
ChicagoInd	0.69523791	0.095849831	7.253408	8.41391E-08	0.498570304	0.89190552
t	-0.596476666	0.113404099	-5.259745	1.51653E-05	-0.82916265	-0.3637907

Example: Hotel Occupancy Case



Much better!! What is the slope of the red line?

Example: Hotel Occupancy Case

Okay, what happened?!

Well, once we account for the downward trend in the occupancy of the plaintiff, the prediction for the occupancy rate is

$$26 + 0.69 * 66 - 0.59 * 31 = 53.25$$

What do we conclude?

Example: Hotel Occupancy Case

Take away lessons...

- When regressing a time series on another, always check the residuals as a time series
- What does that mean... plot the residuals over time. If all is well, you should see no patterns, i.e., they should behave like iid normal samples.

Example: Hotel Occupancy Case

Question

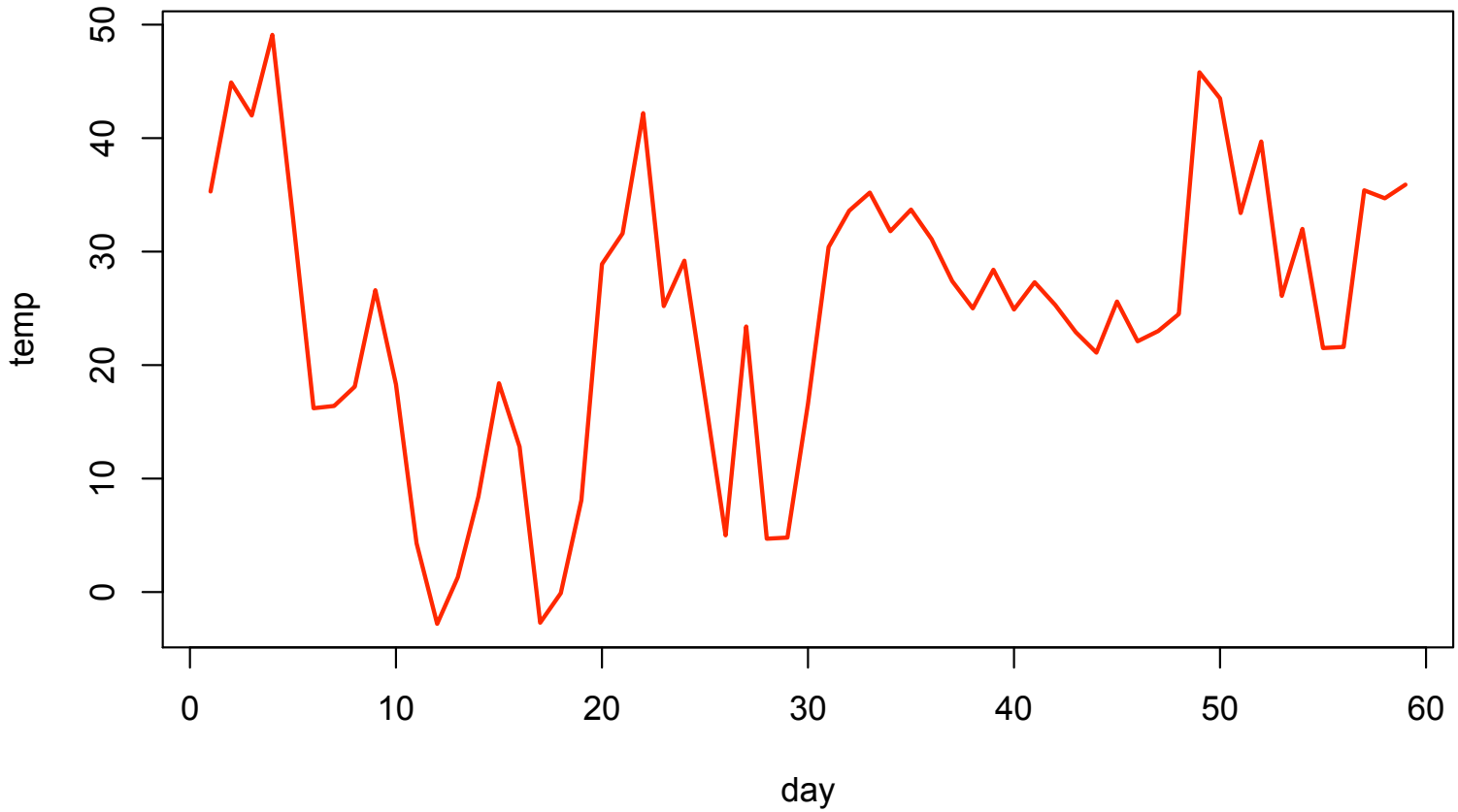
- What if we were interested in predicting the hotel occupancy ten years from now?? We should compute

$$26 + 0.69 * 66 - 0.59 * 150 = -16.96$$

- Would you trust this prediction? Could you defend it in court?
- Remember: always be careful with extrapolating relationships!

Examples: Temperatures

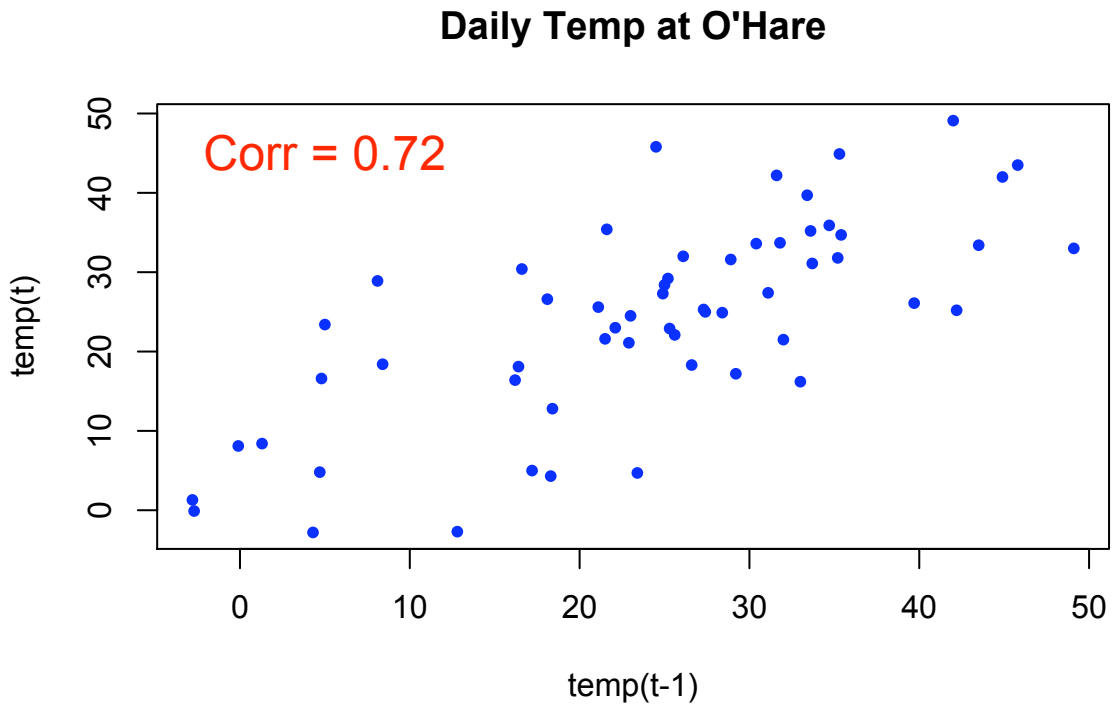
Now you need to predict tomorrow's temperature at O'Hare from (Jan-Feb).



Does this look iid? If it is iid, tomorrow's temperatures should not depend on today's... does that make sense?

Checking for Dependence

To see if Y_{t-1} would be useful for predicting Y_t, just plot them together and see if there is a relationship.



Correlation between Y_t and Y_{t-1} is called autocorrelation.

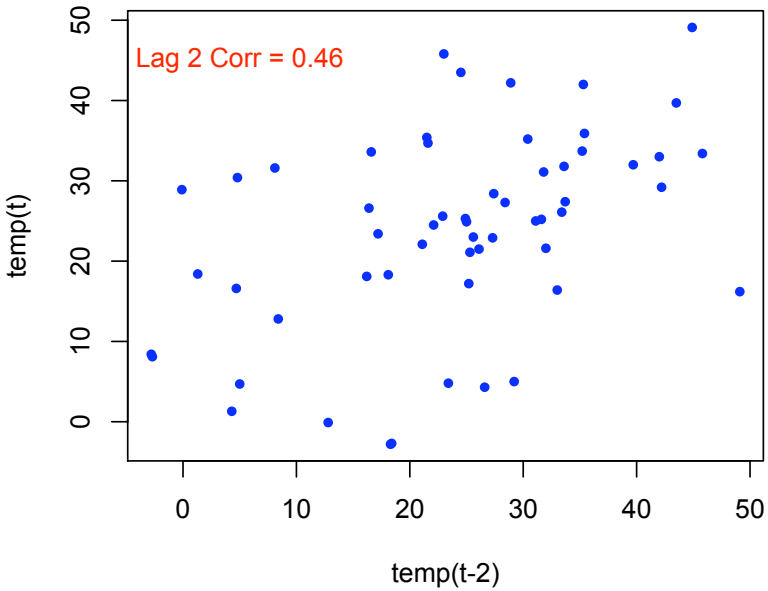
Checking for Dependence

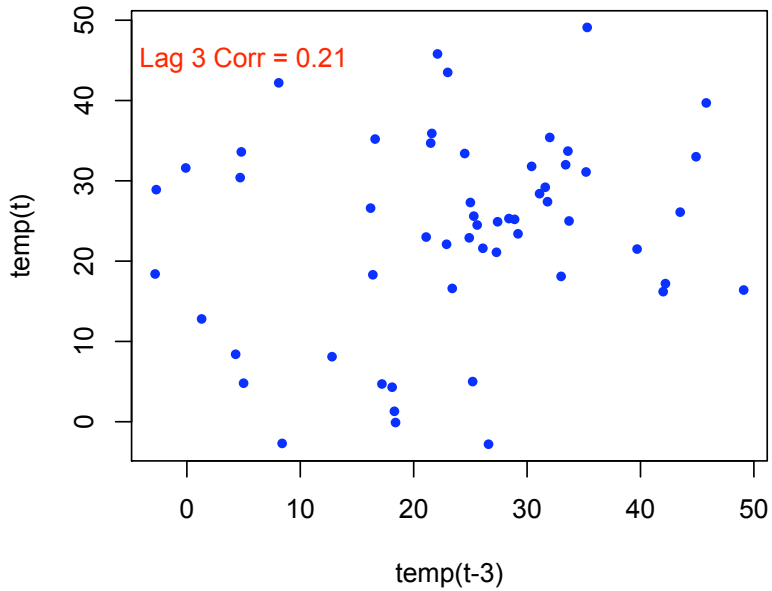
You need to create a "lagged" variable $temp_{t-1}$... the data looks like this:

t	temp(t)	temp(t-1)
1	42	35
2	41	42
3	50	41
4	19	50
5	19	19
6	20	19
...	...	

Checking for Dependence

We can plot Y_t against Y_{t-L} to see L-period lagged relationships.





- It appears that the correlation is getting weaker with increasing L.
- How can we test for this dependence?

The AR(1) Model

A simple way to model dependence over time in with the autoregressive model of order 1...

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \epsilon_t$$

- What is the mean of Y_t for a given value of Y_{t-1}?
- If the model successfully captures the dependence structure in the data then the residuals should look iid.
- Remember: if our data is collected in time, we should always check for dependence in the residuals...

The AR(1) Model

Again, the regression tool is our friend here... (Why?)

SUMMARY OUTPUT

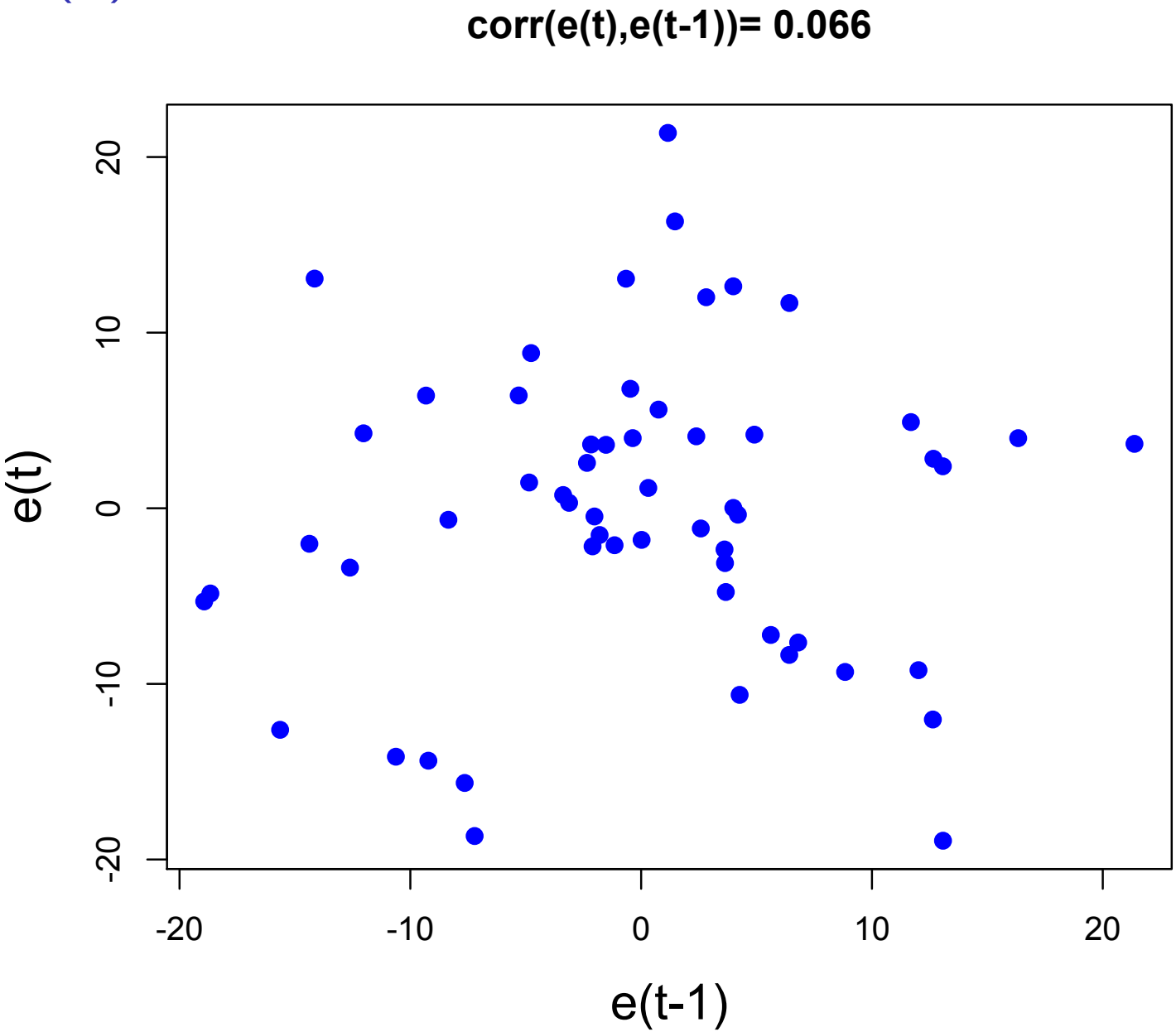
Regression Statistics	
Multiple R	0.722742583
R Square	0.522356842
Adjusted R Sq	0.5138275
Standard Erro	8.789861051
Observations	58

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	1	4731.684433	4731.684433	61.24233673	1.49699E-10
Residual	56	4326.652809	77.2616573		
Total	57	9058.337241			

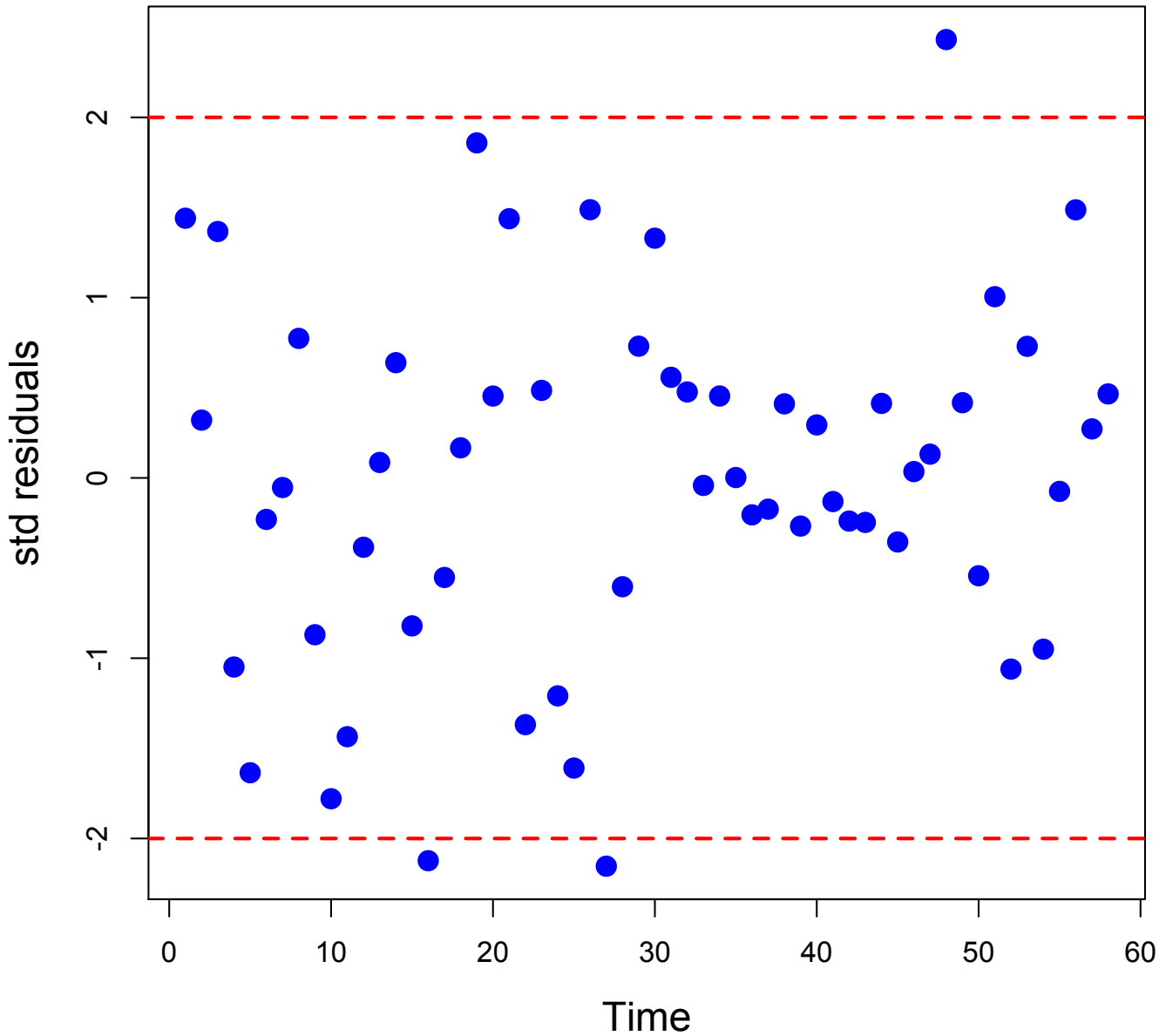
	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	6.705800085	2.516614758	2.664611285	0.010050177	1.664414964	11.74718521
X Variable 1	0.723288866	0.092424243	7.825748317	1.49699E-10	0.53814086	0.908436873

The AR(1) Model



No dependence left!

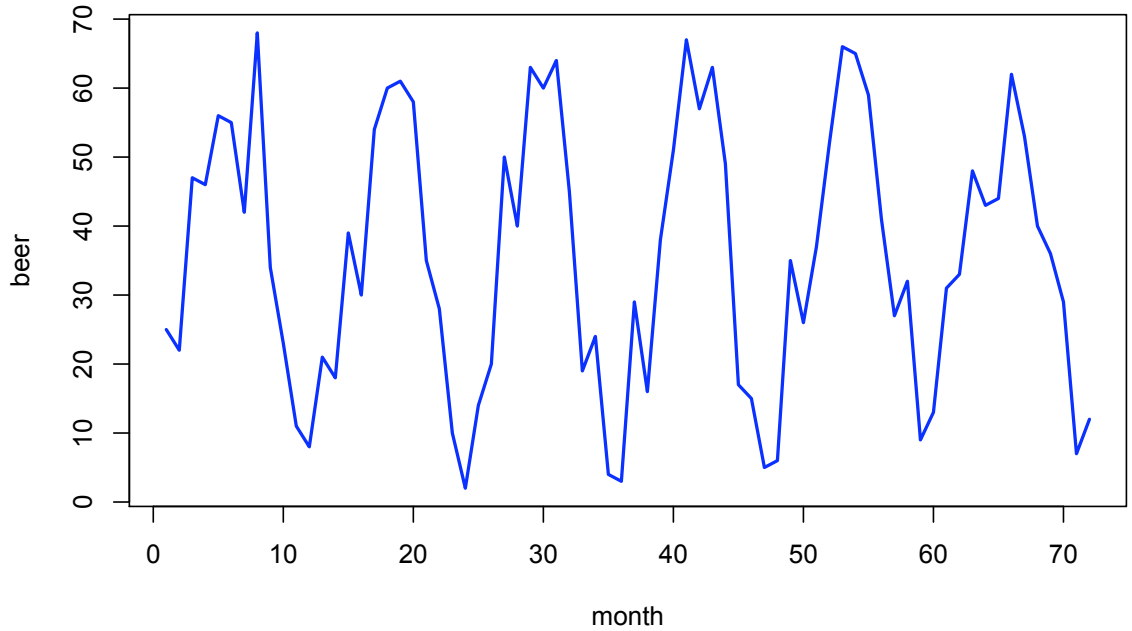
The AR(1) Model



Again, looks good...

The Seasonal Model

- Many time-series data exhibit some sort of seasonality
- The simplest solution is to add a set of dummy variables to deal with the "seasonal effects"



Y_t = monthly U.S. beer production (in millions of barrels).

The Seasonal Model

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.959010553
R Square	0.919701241
Adjusted R Square	0.904979802
Standard Error	0.588667988
Observations	72

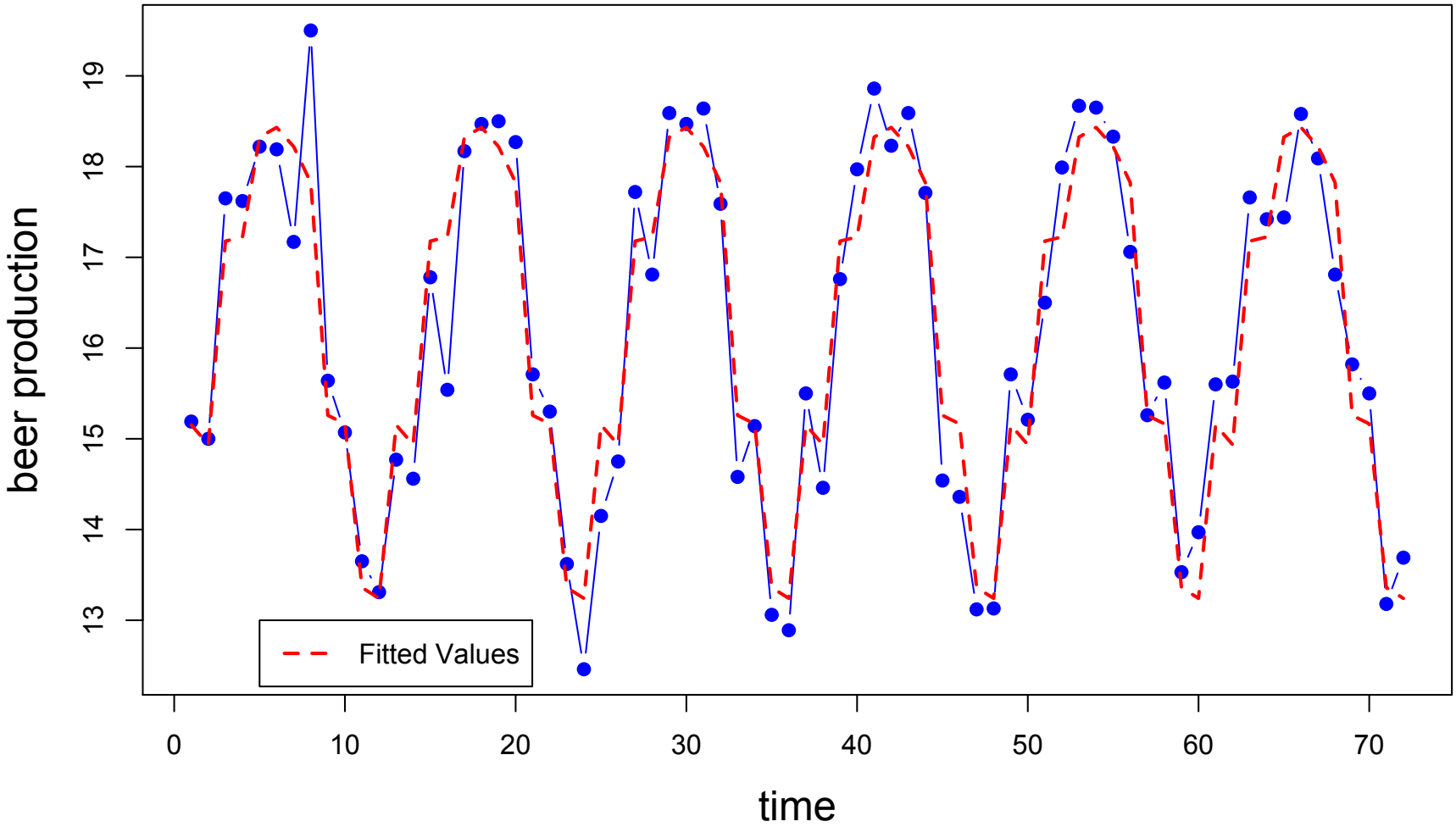
ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	11	238.138728	21.649	62.47359609	1.20595E-28
Residual	60	20.7918	0.34653		
Total	71	258.930528			

	Coefficients	*Standard Erro*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*
Intercept	13.24166667	0.2403227	55.0995	4.32368E-53	12.7609497	13.72238
X Variable 1	1.911666667	0.33986762	5.62474	5.15088E-07	1.23183021	2.591503
X Variable 2	1.693333333	0.33986762	4.98233	5.64079E-06	1.013496877	2.37317
X Variable 3	3.936666667	0.33986762	11.5829	6.13313E-17	3.25683021	4.616503
X Variable 4	3.983333333	0.33986762	11.7202	3.74305E-17	3.303496877	4.66317
X Variable 5	5.083333333	0.33986762	14.9568	6.59589E-22	4.403496877	5.76317
X Variable 6	5.19	0.33986762	15.2707	2.44866E-22	4.510163543	5.869836
X Variable 7	4.978333333	0.33986762	14.6479	1.77048E-21	4.298496877	5.65817
X Variable 8	4.581666667	0.33986762	13.4807	8.22861E-20	3.90183021	5.261503
X Variable 9	2.016666667	0.33986762	5.93368	1.58522E-07	1.33683021	2.696503
X Variable 10	1.923333333	0.33986762	5.65907	4.52211E-07	1.243496877	2.60317
X Variable 11	0.118333333	0.33986762	0.34817	0.728927584	-0.561503123	0.79817

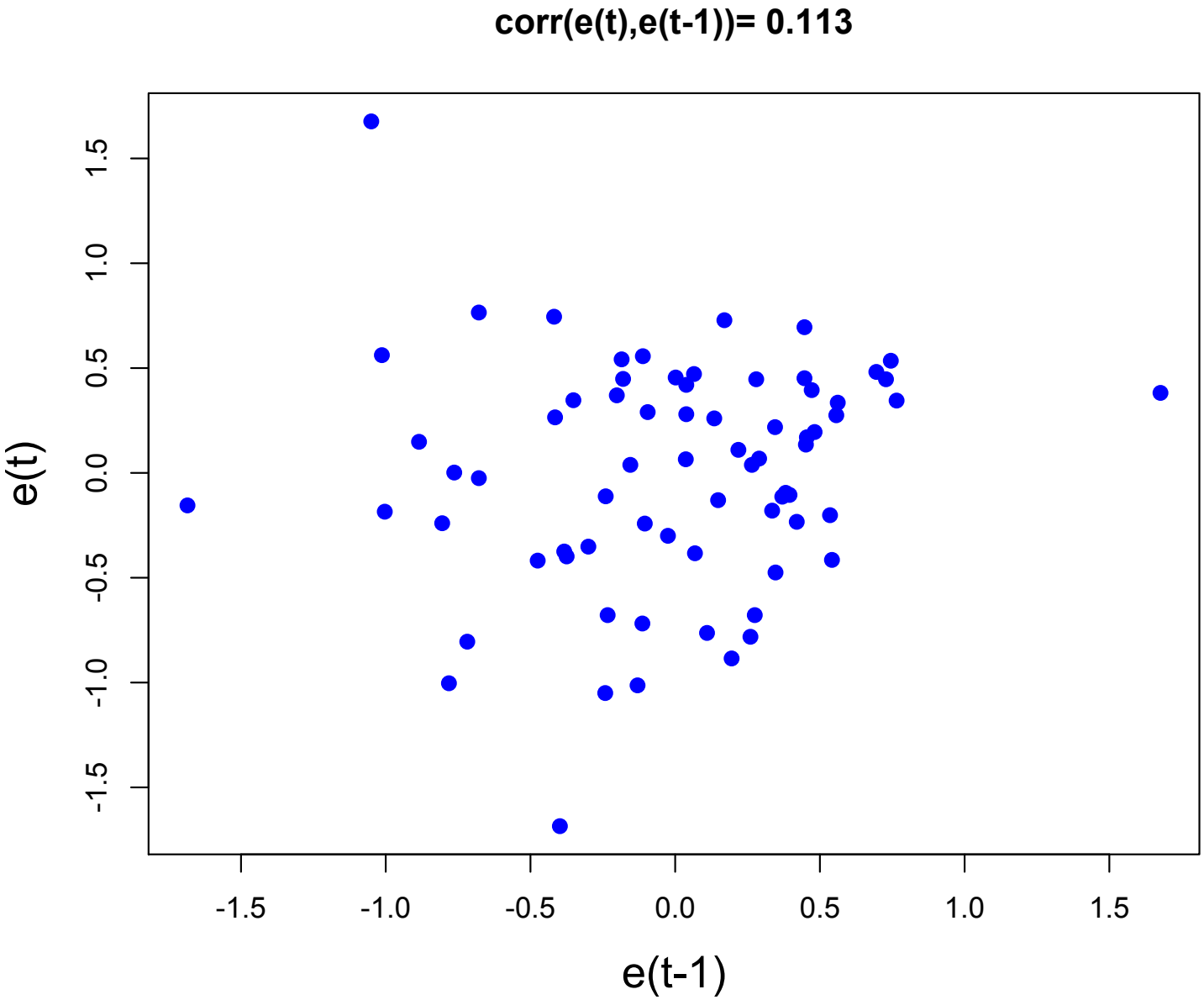
Let's look at the Excel file...

The Seasonal Model



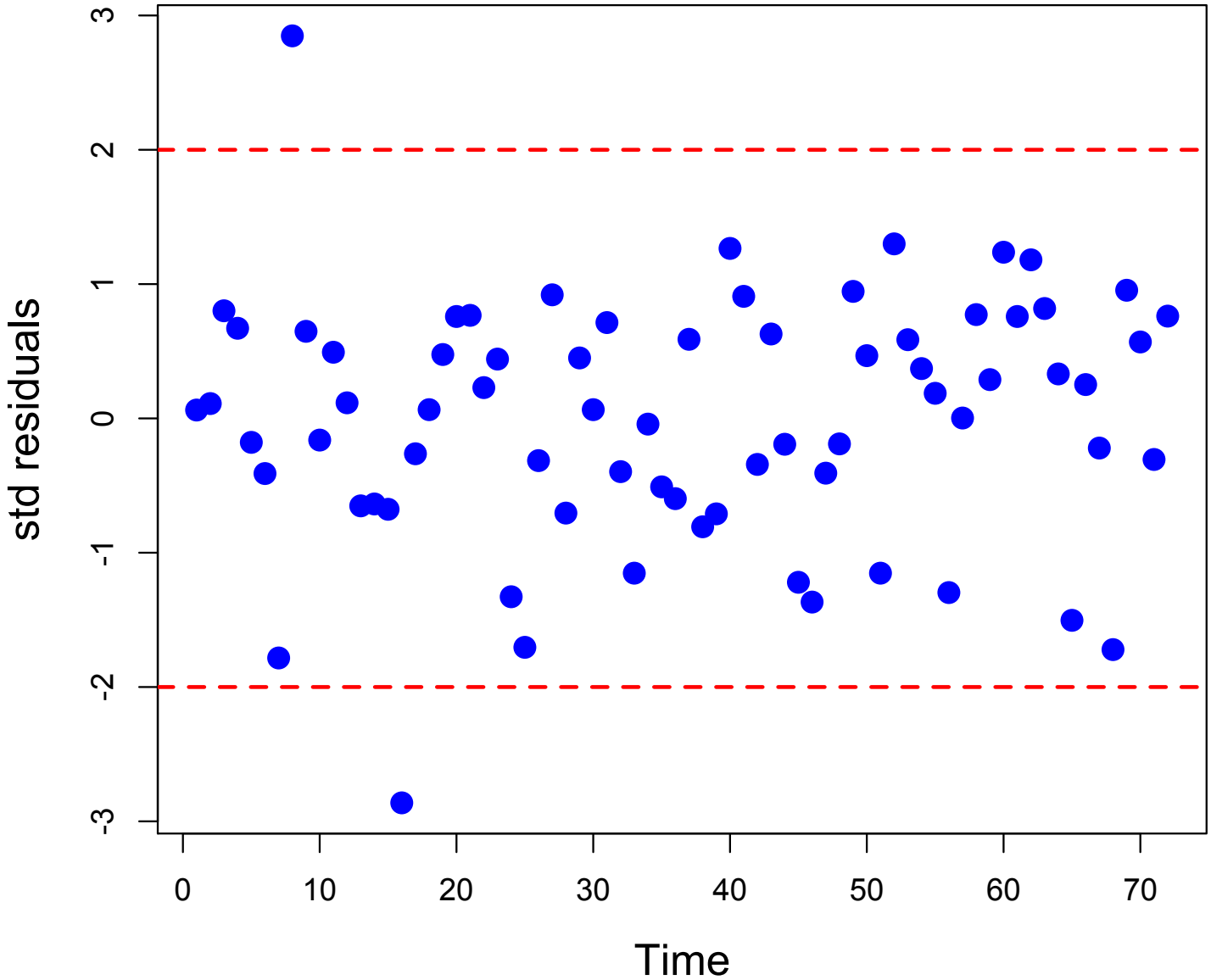
What would our future predictions look like?

The Seasonal Model



Okay... good enough.

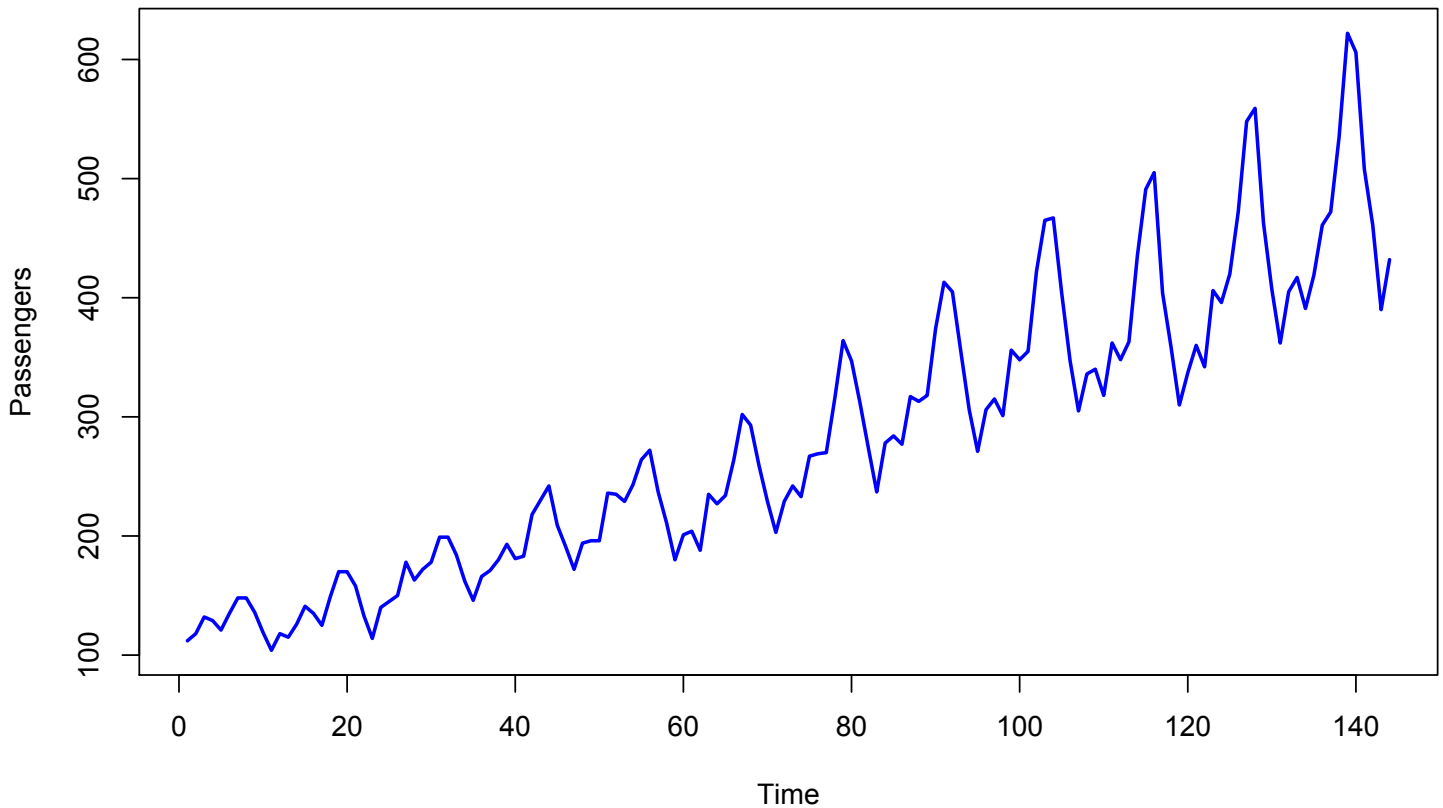
The Seasonal Model



Still, no obvious problems...

Airline Data

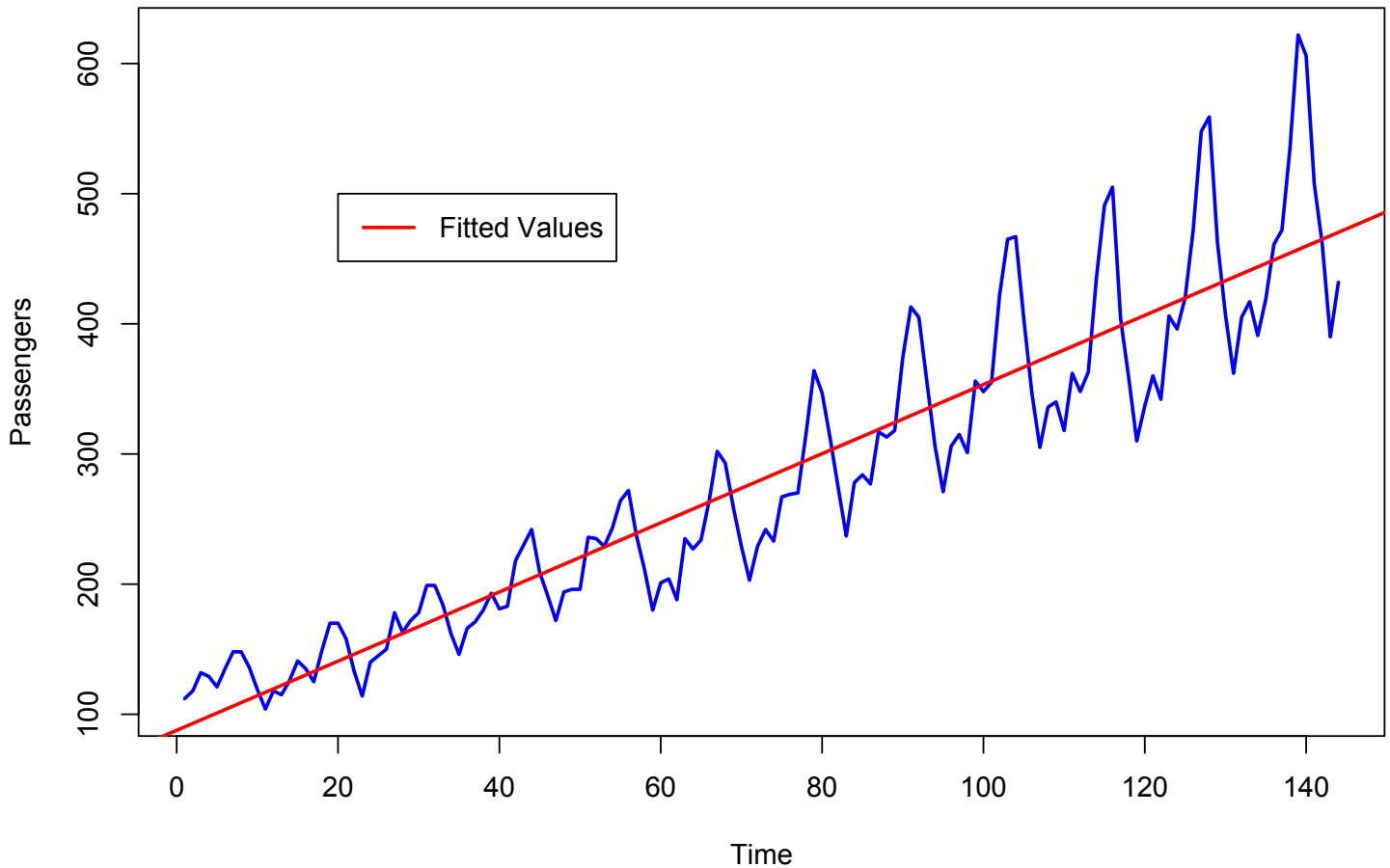
Monthly passengers in the U.S. airline industry (in 1,000 of passengers) from 1949 to 1960... we need to predict the number of passengers in the next couple of months.



Any ideas?

Airline Data

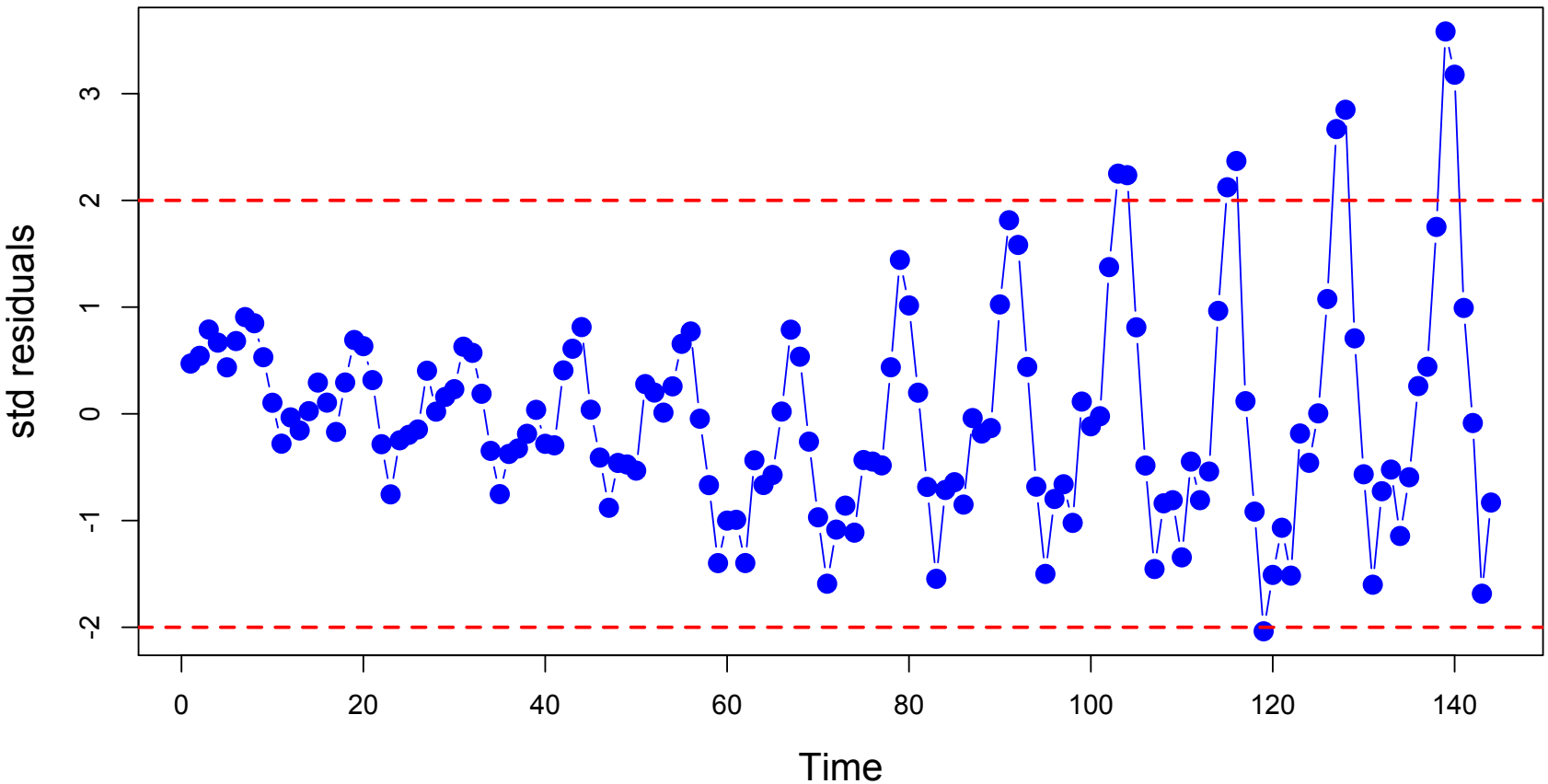
How about a "trend model"? $Y_t = \beta_0 + \beta_1 t + \epsilon_t$



What do you think?

Airline Data

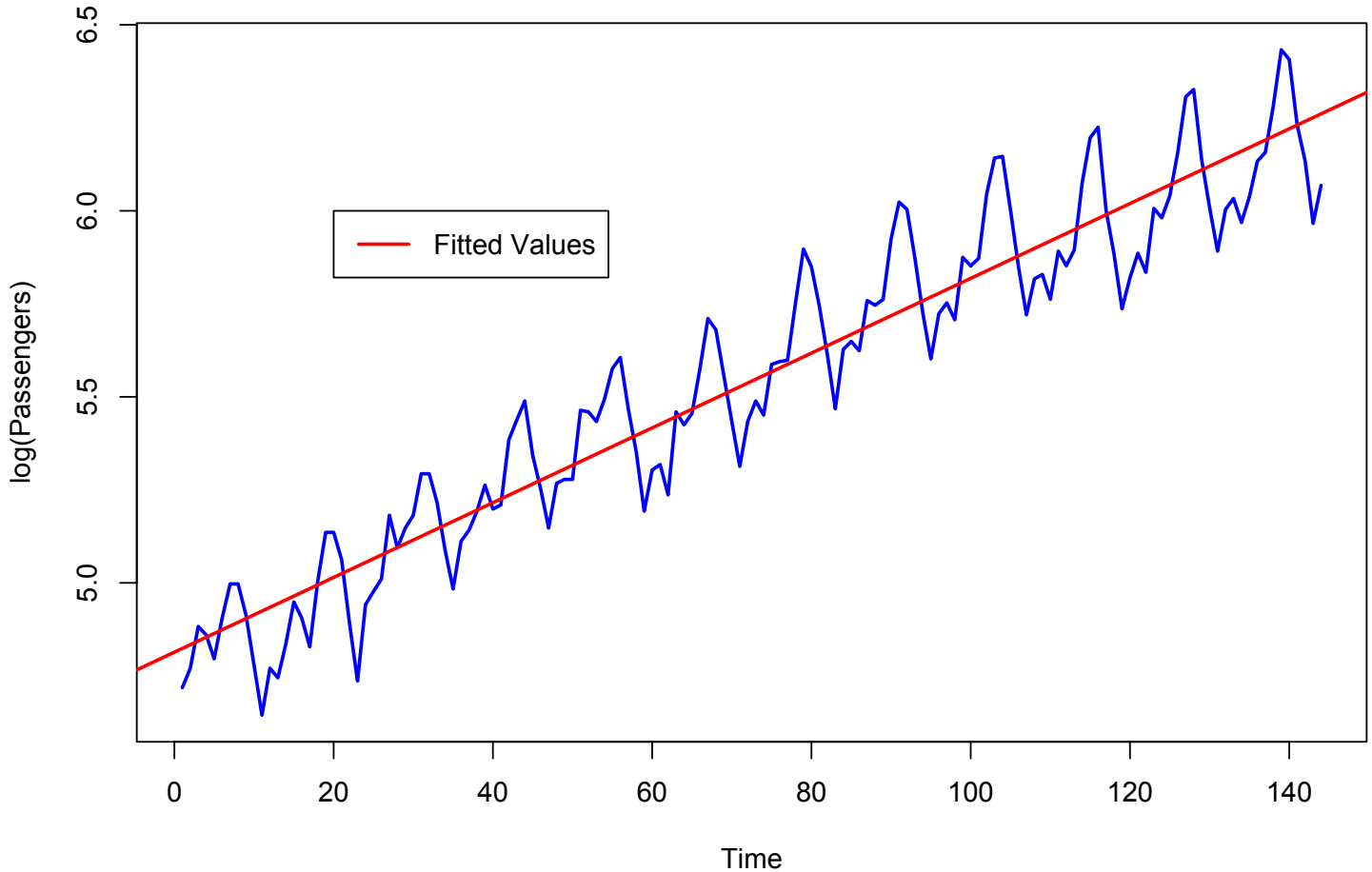
Let's look at the residuals...



Is there any obvious pattern here? YES!!

Airline Data

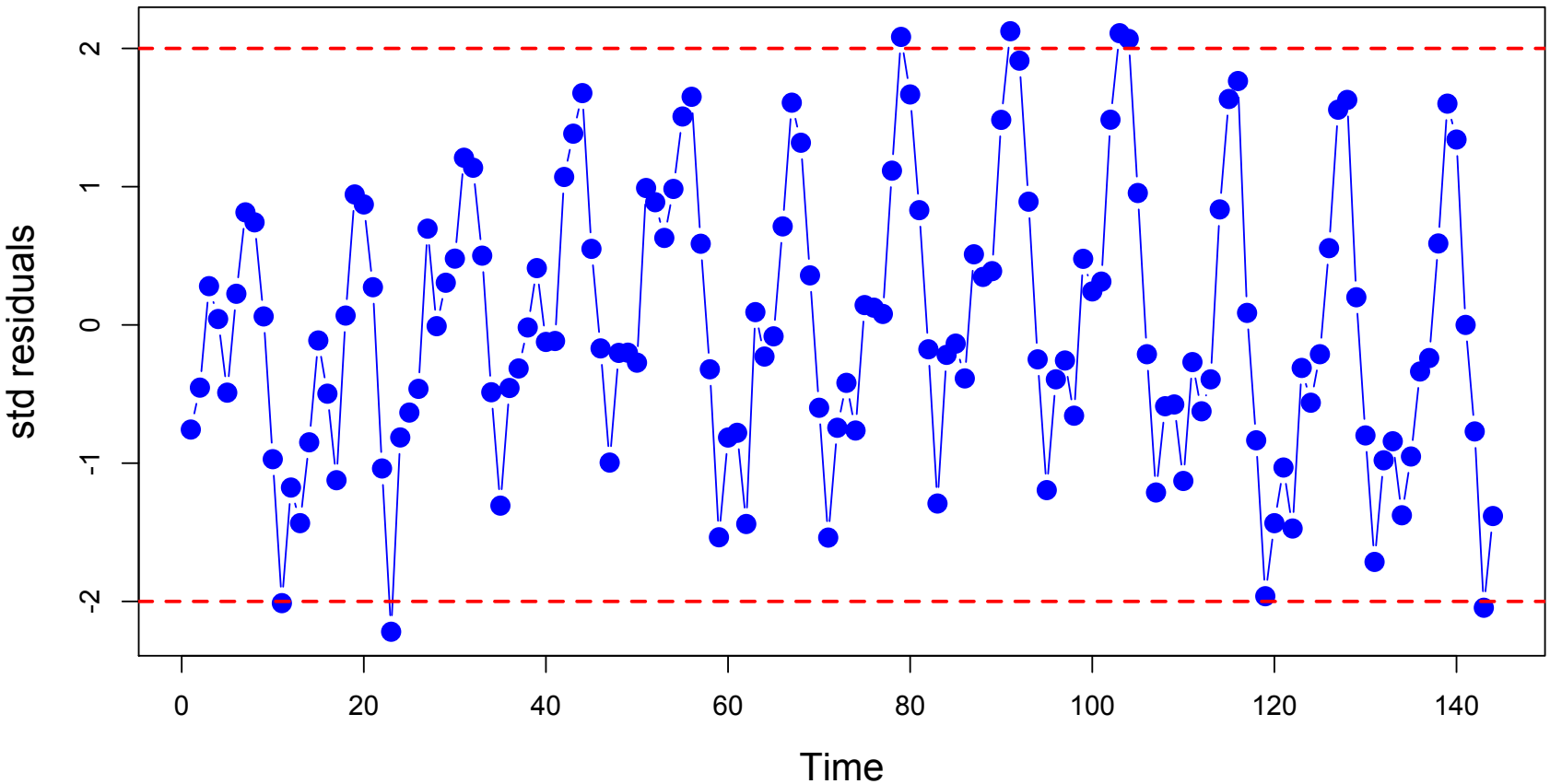
The variance of the residuals seems to be growing in time... Let's try taking the log. $\log(Y_t) = \beta_0 + \beta_1 t + \epsilon_t$



Any better?

Airline Data

Residuals...

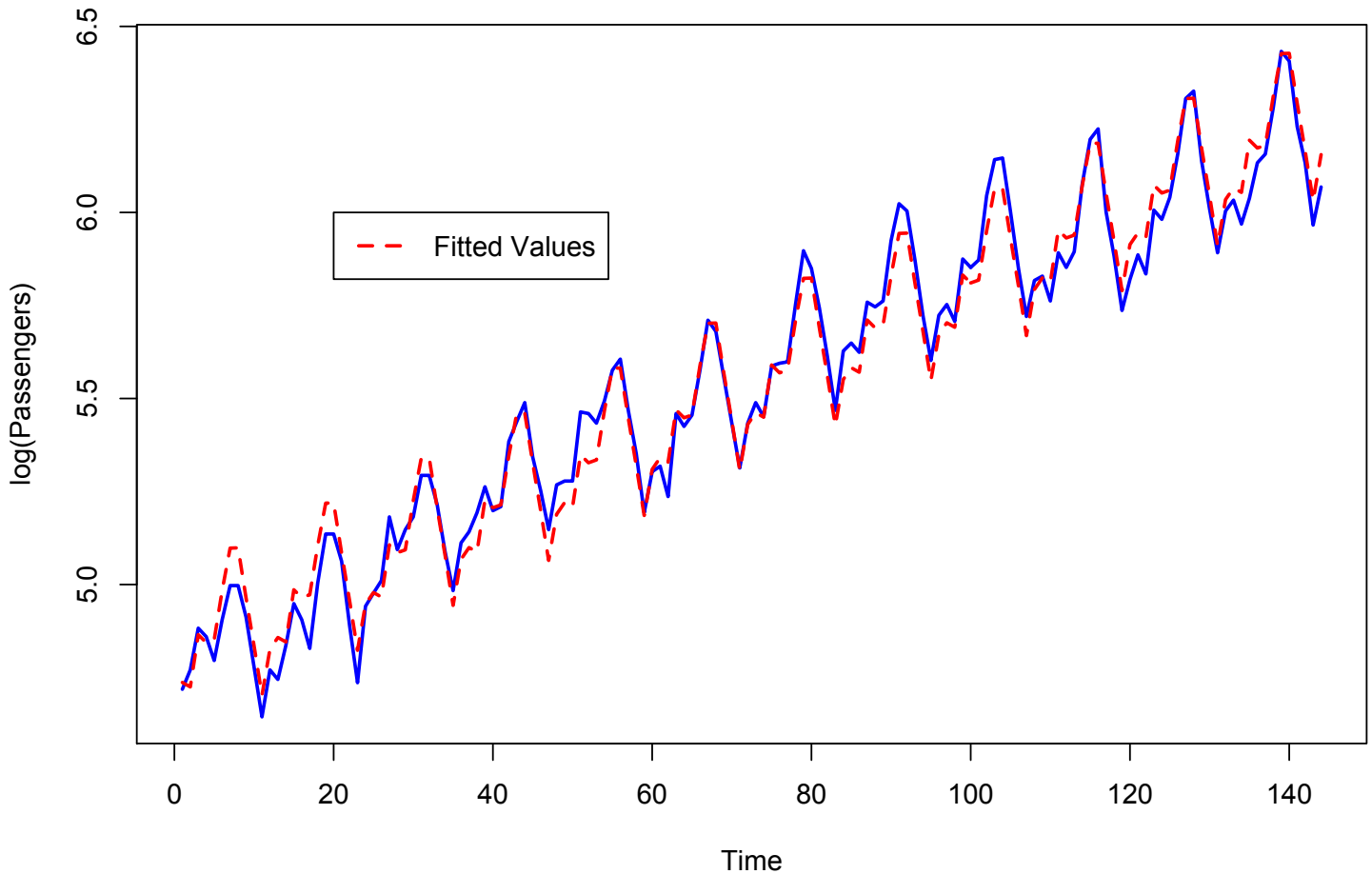


Still we can see some obvious temporal/seasonal pattern....

Airline Data

Okay, let's add dummy variables for months (only 11 dummies)...

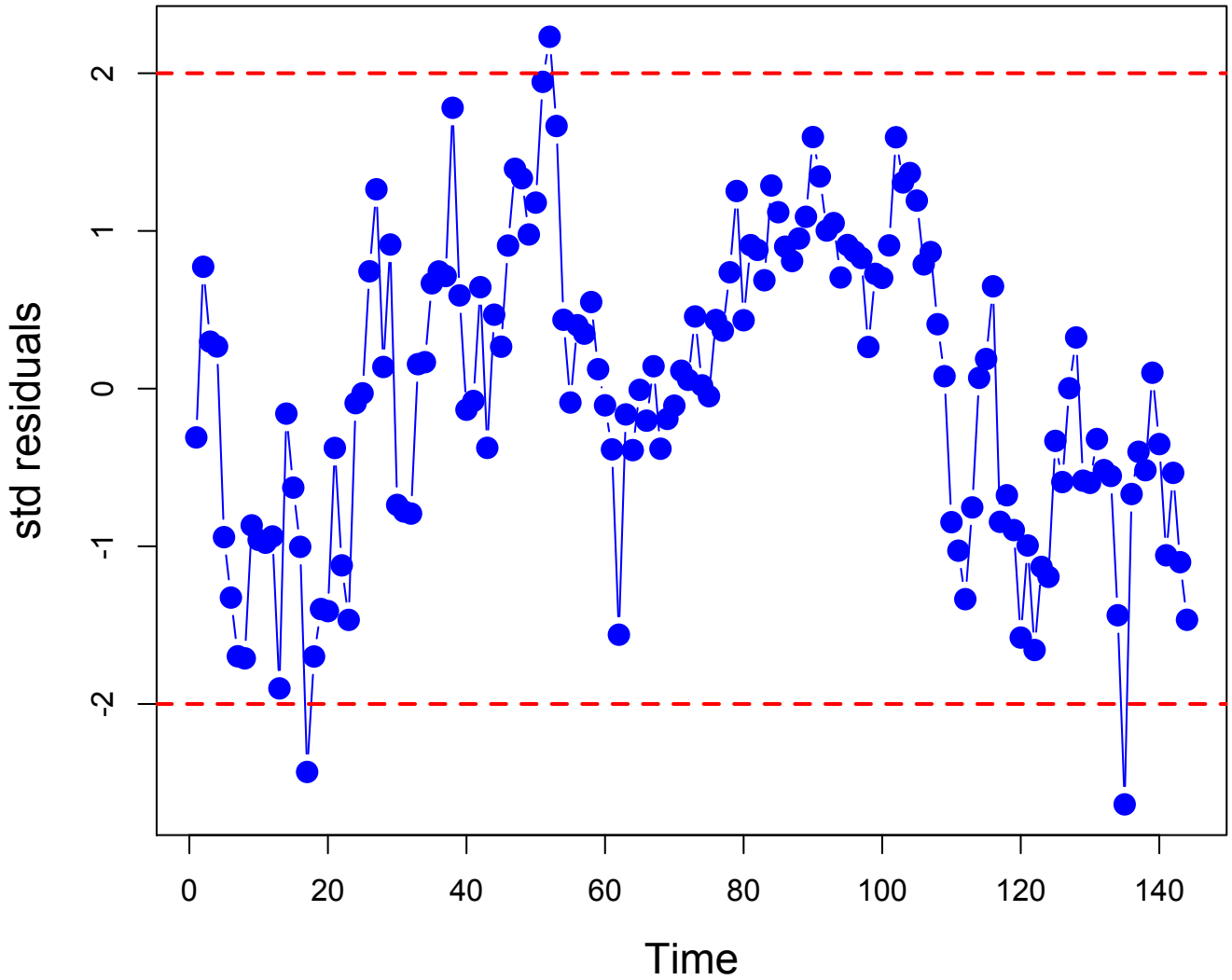
$$\log(Y_t) = \beta_0 + \beta_1 t + \beta_2 Jan + ... \beta_{12} Dec + \epsilon_t$$



Much better!!

Airline Data

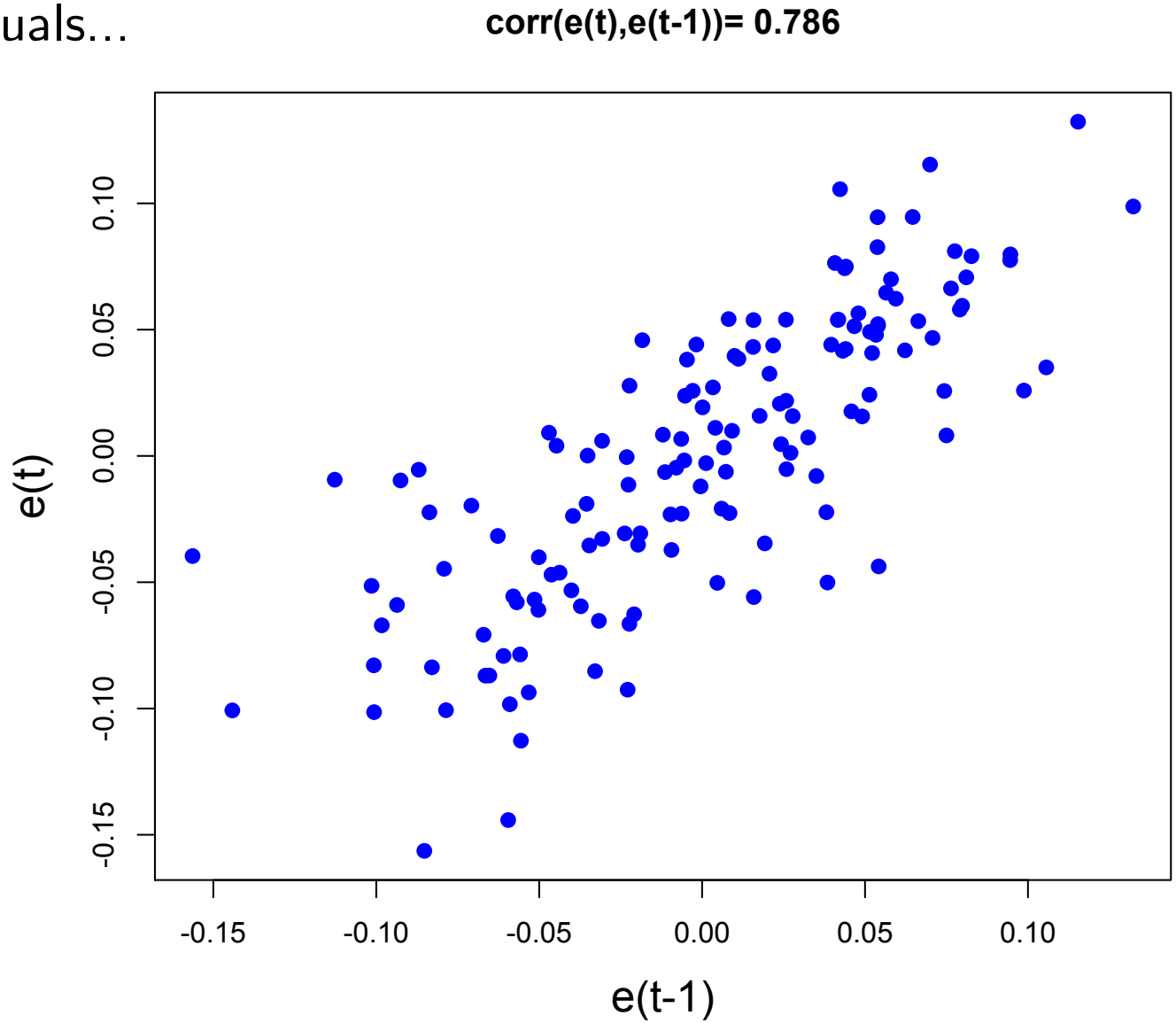
Residuals...



I am still not happy... it doesn't look normal iid to me...

Airline Data

Residuals...

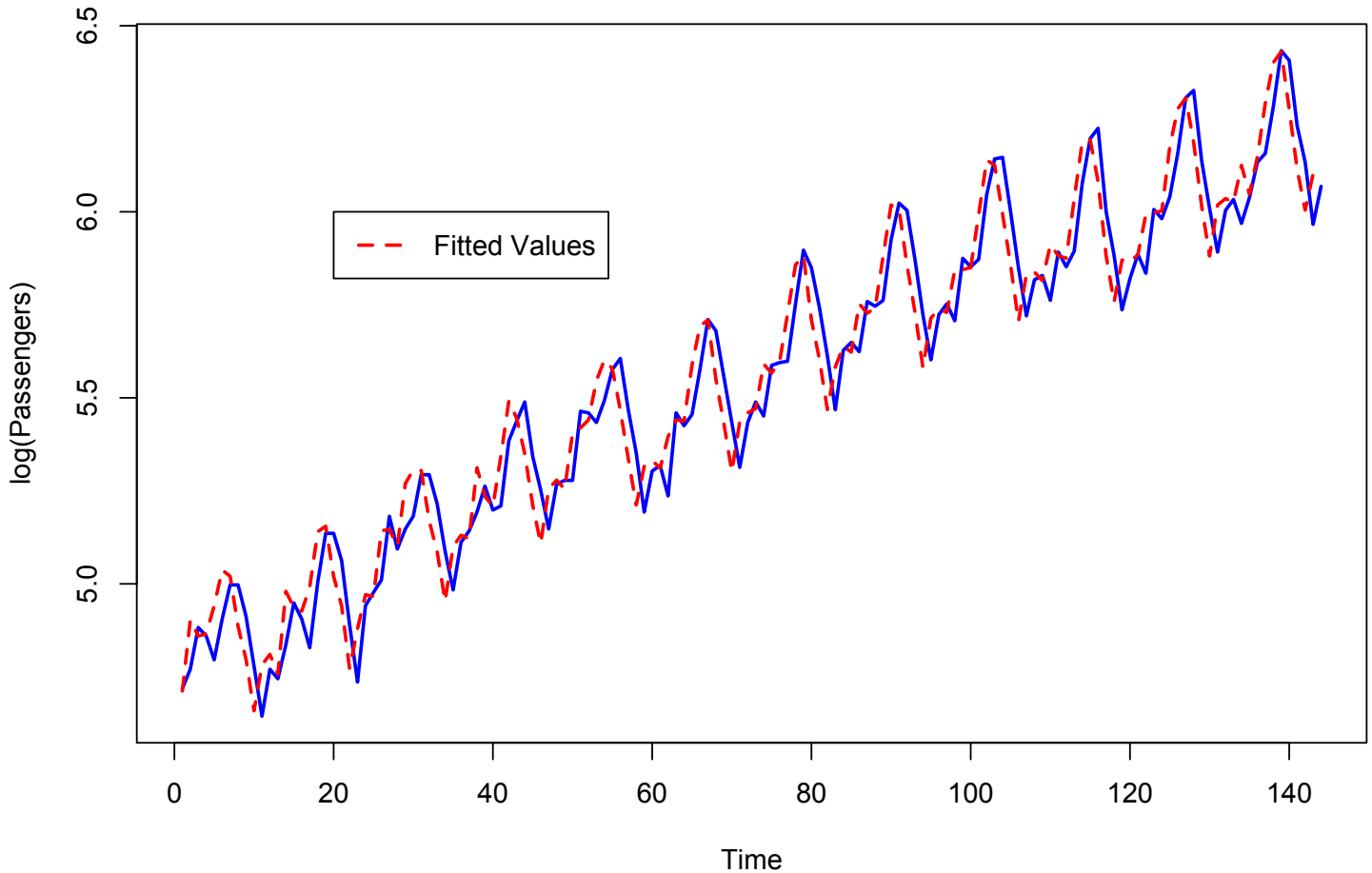


I was right! The residuals are dependent on time...

Airline Data

We have one more tool... let's add one legged term.

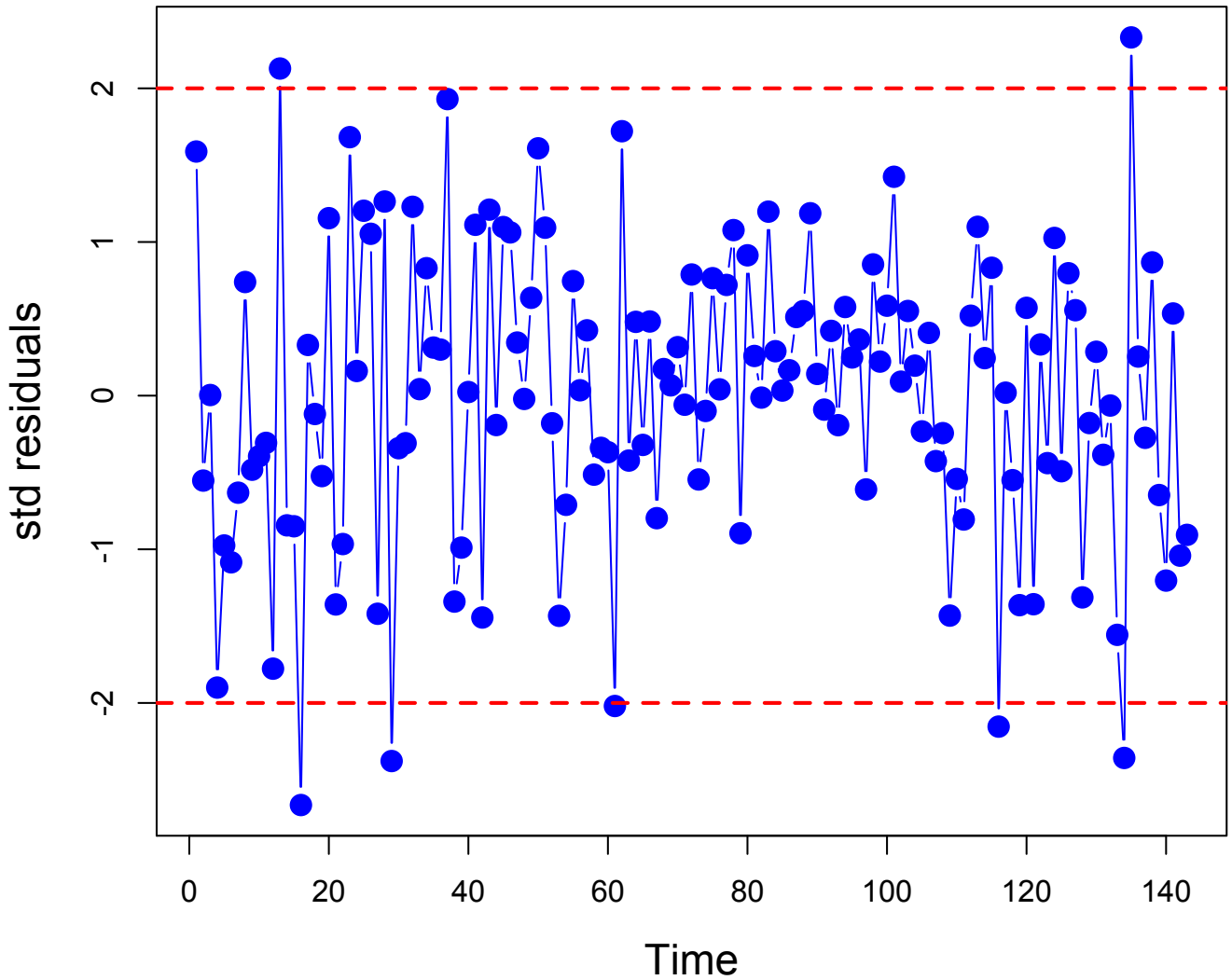
$$\log(Y_t) = \beta_0 + \beta_1 t + \beta_2 Jan + ...\beta_{12} Dec + \beta_{13} \log(Y_{t-1}) + \epsilon_t$$



Okay, good...

Airline Data

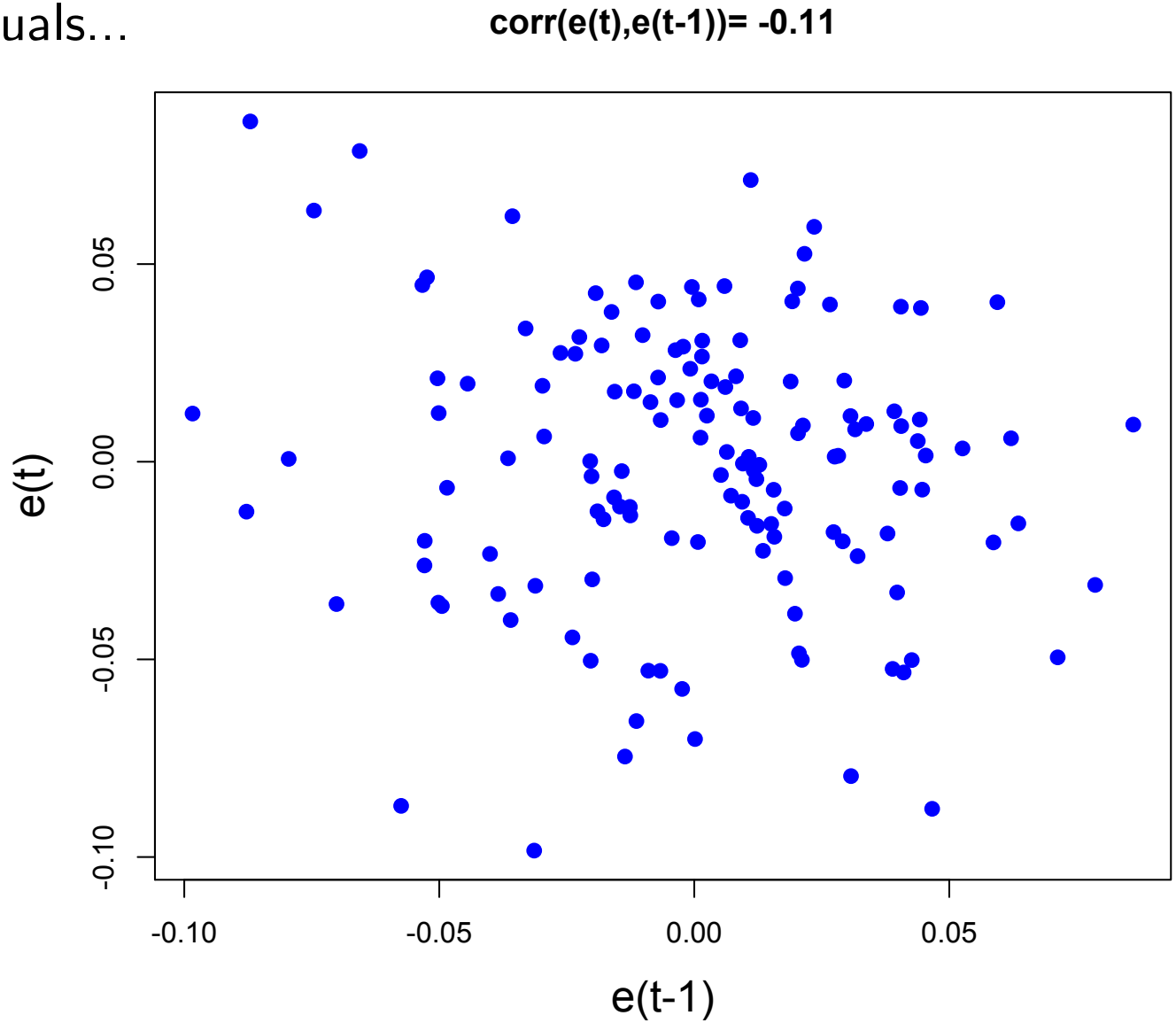
Residuals...



Much better!!

Airline Data

Residuals...



Much better indeed!!

Summary

Whenever working with time series data we need to look for dependencies over time.

We can deal with lots of types of dependencies by using regression models... our tools are:

- trends
- lags
- seasonal dummies

Binary Response Data

Let's now look at data where the response Y is a binary variable (taking the value 0 or 1).

- Win or lose.
- Sick or healthy.
- Buy or not buy.
- Pay or default.
- Thumbs up or down.

The goal is generally to predict the probability that $Y = 1$, and you can then do classification based on this estimate.

Binary Response Data

Y is an indicator: $Y = 0$ or 1. The conditional mean is thus

$$\mathbb{E}[Y|X] = p(Y = 1|X) \times 1 + p(Y = 0|X) \times 0 = p(Y = 1|X)$$

The mean function is a probability: We need a model that gives mean/probability values between 0 and 1.

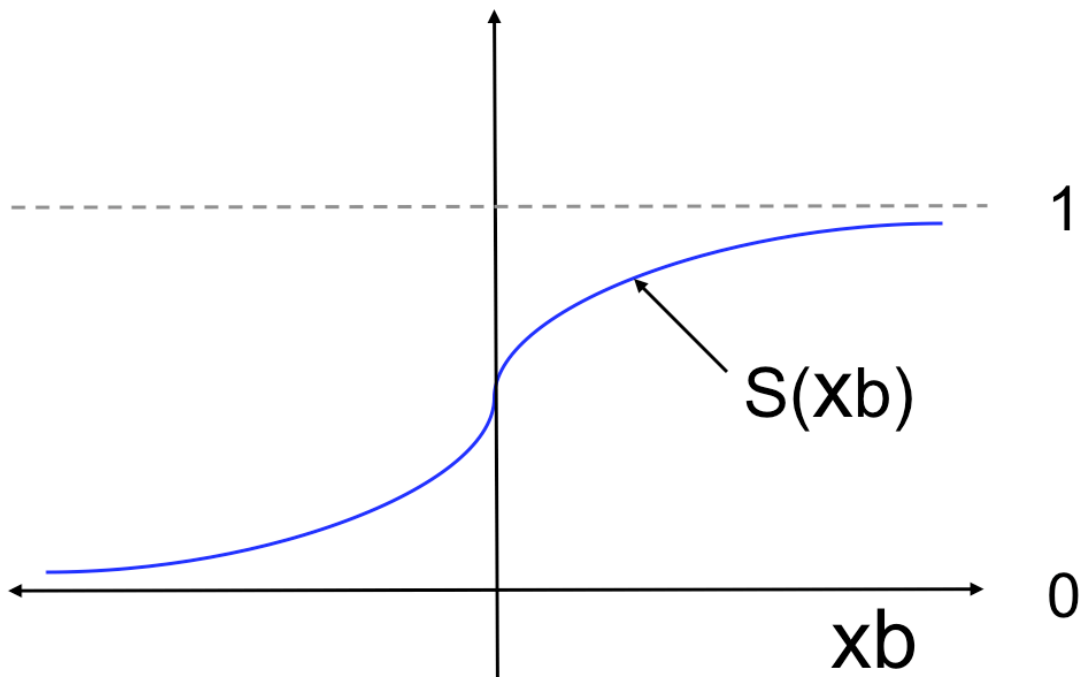
We'll use a transform function that takes the right-hand side of the model ($\mathbf{x}'\beta$) and gives back a value between zero and one.

Binary Response Data

The binary choice model is

$$p(Y = 1|X_1 \ldots X_d) = S(\beta_0 + \beta_1 X_1 \ldots + \beta_d X_d)$$

where S is a function that increases in value from zero to one.



Binary Response Data

There are two main functions that are used for this:

- Logistic Regression: $S(z) = \dfrac{e^z}{1+e^z}$.
- Probit Regression: $S(z) = \texttt{pnorm}(z)$.

Both functions are S-shaped and take values in $(0, 1)$.

Probit is used by economists, logit by biologists, and the rest of us are fairly indifferent: they result in practically the same fit.

Logistic Regression

We'll use logistic regression, such that

$$p(Y = 1|X_1 \dots X_d) = \frac{\exp[\beta_0 + \beta_1 X_1 \dots + \beta_d X_d]}{1 + \exp[\beta_0 + \beta_1 X_1 \dots + \beta_d X_d]}$$

The "logit" link is more common, and it's the default in R.

These models are easy to fit in R:

```
glm(Y ~ X1 + X2, family=binomial)
```

"g" stands for generalized, and `binomial` indicates $Y = 0$ or 1.

Otherwise, generalized linear models use the same syntax as lm().

Logistic Regression

What is happening here? Instead of least-squares,

`glm` is maximizing the product of probabilities:

$$\prod_{i=1}^{n} P(Y_i|\mathbf{x}_i) = \prod_{i=1}^{n} \left(\frac{\exp[\mathbf{x}'\mathbf{b}]}{1+\exp[\mathbf{x}'\mathbf{b}]}\right)^{Y_i} \left(\frac{1}{1+\exp[\mathbf{x}'\mathbf{b}]}\right)^{1-Y_i}$$

This maximizes the likelihood of our data

(which is also what least-squares did).

Logistic Regression

The important things are basically the same as before:

- Individual parameter p-values are interpreted as always.
- `extractAIC(reg,k=log(n))` will get your BICs.
- The `predict` function works as before, but you need to add `type = ''response''` to get $\hat{p}_i = \exp[\mathbf{x}'b]/(1+\exp[\mathbf{x}'b])$ (otherwise it just returns the linear function $\mathbf{x}'\beta$).

Unfortunately, techniques for residual diagnostics and model checking are different (but we'll not worry about that today).

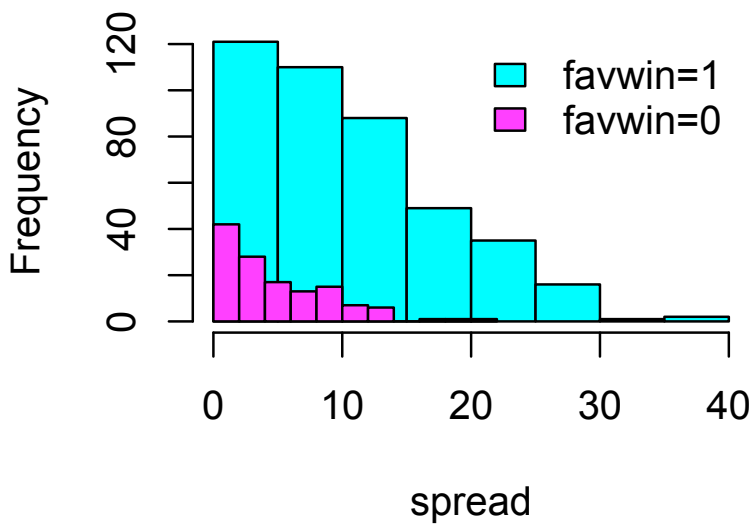
Also, without sums of squares there are no R^2, anova, or F-tests!

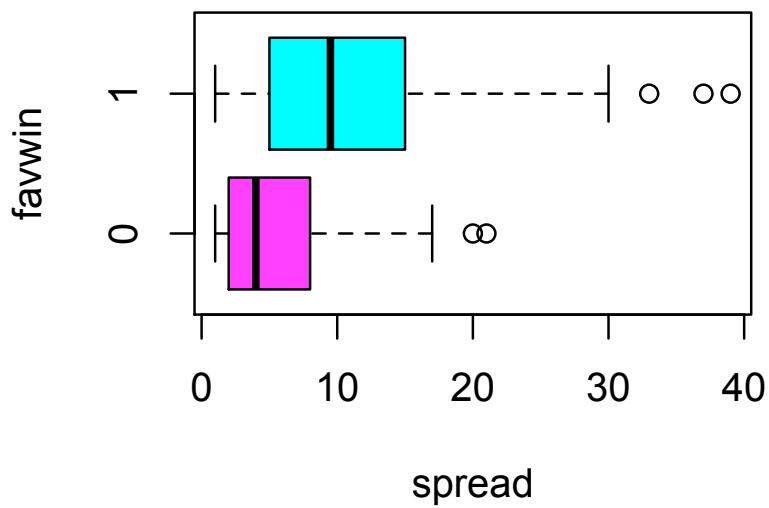
Example: Basketball Spreads

NBA basketball point spreads: we have Las Vegas betting point spreads for 553 NBA games and the resulting scores.

We can use logistic regression of scores onto spread to predict the probability of the favored team winning.

- Response: `favwin=1` if favored team wins.
- Covariate: `spread` is the Vegas point spread.





Example: Basketball Spreads

This is a weird situation where we assume is no intercept.

- There is considerable evidence that betting odds are efficient.
- A spread of zero implies $p(win) = 0.5$ for each team.
- Thus $p(win) = \exp[\beta_0]/(1 + \exp[\beta_0]) = 1/2 \Leftrightarrow \beta_0 = 0$.

The model we want to fit is thus

$$p(favwin|spread) = \frac{\exp[\beta \times spread]}{1 + \exp[\beta \times spread]}$$

Example: Basketball Spreads

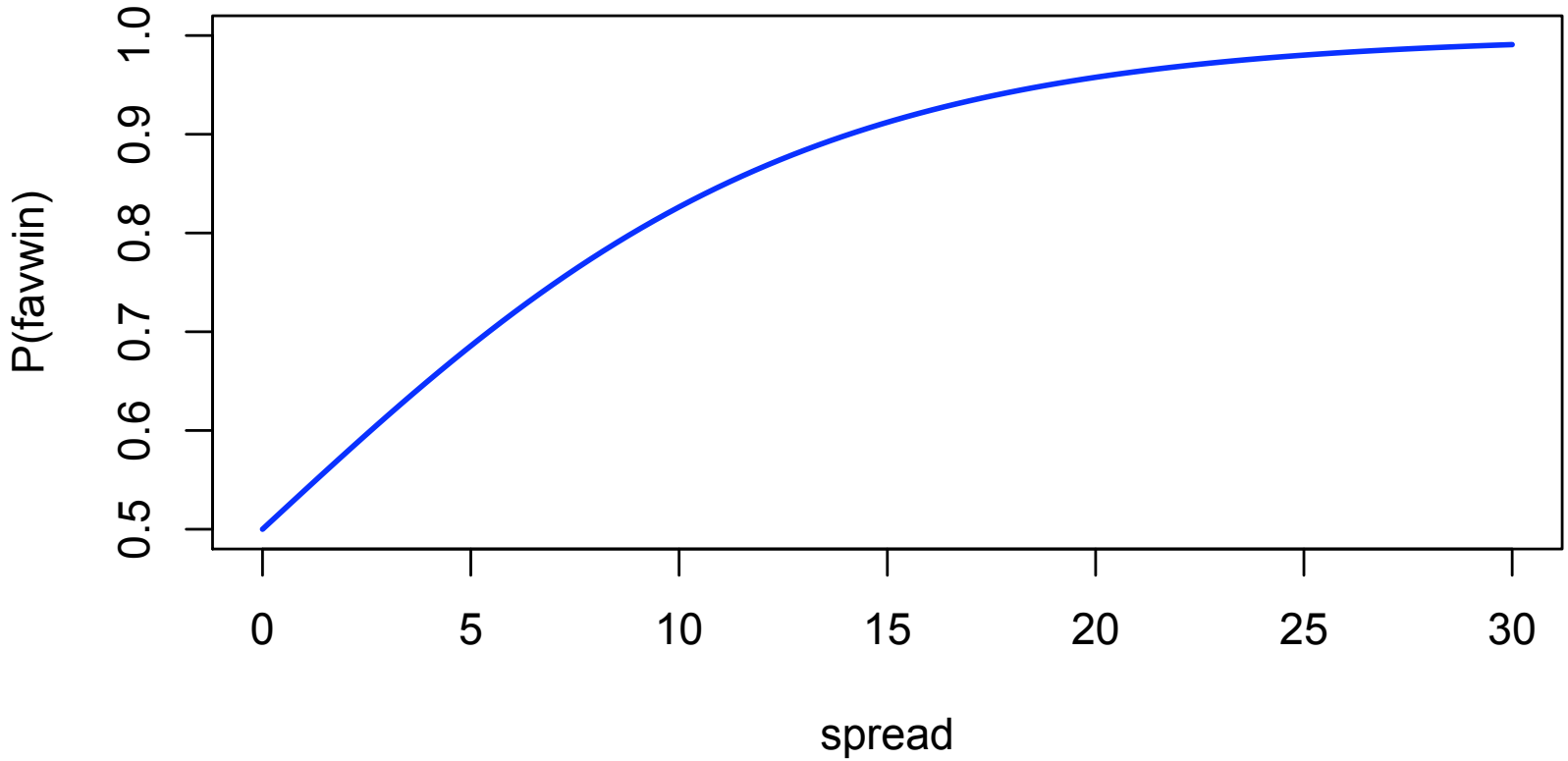
```
summary(nbareg <- glm(favwin ~ spread-1, family=binomial))

Call:
glm(formula = favwin ~ spread - 1, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5741   0.1587   0.4619   0.8135   1.1119

Coefficients:
       Estimate Std. Error z value Pr(>|z|)
spread  0.15600    0.01377   11.33   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 766.62  on 553  degrees of freedom
Residual deviance: 527.97  on 552  degrees of freedom
AIC: 529.97

Number of Fisher Scoring iterations: 5
```

Some things are different (z not t) and some are missing (F, R^2).

Example: Basketball Spreads

The fitted model is

$$p(favwin|spread) = \frac{\exp[0.156 \times spread]}{1 + \exp[0.156 \times spread]}$$



Example: Basketball Spreads

We could consider other models... and compare with BIC!

Our "Efficient Vegas" model:

```
> extractAIC(nbareg, k=log(553))
   1.000 534.287
```

A model that includes non-zero intercept:

```
> extractAIC(glm(favwin ~ spread, family=binomial), k=log(553))
   2.0000 540.4333
```

What if we throw in home-court advantage?

```
> extractAIC(glm(favwin ~ spread+favhome, family=binomial), k=log(553))
   3.0000 545.6371
```

The simplest model is best
(The model probabilities are 19/20, 1/20, and zero.)

Example: Basketball Spreads

Let's use our model to predict the result of a game:

- Portland vs Golden State: spread is PRT by 8

$$p(PRT\ win) = \frac{\exp[0.156 \times 8]}{1 + \exp[0.156 \times 8]} = 0.78$$

- Chicago vs Orlando: spread is ORL by 4

$$p(CHI\ win) = \frac{1}{1 + \exp[0.156 \times 4]} = 0.35$$

Example: Credit Scoring

A common business application of logistic regression is in evaluating the credit quality of (potential) debtors.

- Take a list of borrower characteristics.
- Build a prediction rule for their credit.
- Use this rule to automatically evaluate applicants (and track your risk profile).

You can do all this with logistic regression, and then use the predicted probabilities to build a classification rule.

Example: Credit Scoring

We have data on 1000 loan applicants at German community banks, and judgement of the loan outcomes (good or bad).

The data has 20 borrower characteristics, including

- Credit history (5 categories).
- Housing (rent, own, or free).
- The loan purpose and duration.
- Installment rate as a percent of income.

Example: Credit Scoring

We can use forward step wise regression to build a model.

```
null <- glm(Y ~ history3, family=binomial, data=credit[train,])
full <- glm(Y ~., family=binomial, data=credit[train,])
reg <- step(null, scope=formula(full), direction="forward", k=log(n))
```

⋮

```
Step:  AIC=882.94

Y[train] ~ history3 + checkingstatus1 + duration2 + installment8
```

The null model has credit history as a variable, since I'd include this regardless, and we've left-out 200 points for validation.

Classification

A common goal with logistic regression is to classify the inputs depending on their predicted response probabilities.

For example, we might want to classify the German borrowers as having "good" or "bad" credit (i.e., do we loan to them?).

A simple classification rule is to say that anyone with $p(\mathit{good}|\mathbf{x}) > 0.5$ can get a loan, and the rest do not.

Example: Credit Scoring

Let's use the validation set to compare this and the full model.

```
> full <- glm(formula(terms(Y[train] ~., data=covars)),
                  data=covars[train,], family=binomial)
> predreg <- predict(reg, newdata=covars[-train,], type="response")
> predfull <- predict(full, newdata=covars[-train,], type="response")
> # 1 = false negative, -1 = false positive
> errorreg <- Y[-train]-(predreg >= .5)
> errorfull <- Y[-train]-(predfull >= .5)
> # misclassification rates:
> mean(abs(errorreg))
    0.220
> mean(abs(errorfull))
    0.265
```

Our model classifies borrowers correctly 78% of the time.

Classification

You can also do classification with cut-offs other than $1/2$.

- Suppose the risk associated with one action is higher than for the other.
- You'll want to have $p > 0.5$ of a positive outcome before taking the risky action.

k-Nearest Neighbors (kNN)

The k-nearest neighbors algorithm will try to *predict* (numerical variables) or *classify* (categorical variables) based on similar records on the *training dataset*.

Remember, the problem is to guess a future value y_f given new values of the covariates $X_f = (x_{1f}, x_{2f}, x_{3f}, \ldots, x_{pf})$.

k-Nearest Neighbors (kNN)

kNN: How do the $y's$ look like close to the region around X_f?

We need to find the k records in the training dataset that are close to X_f. How? "Nearness" to the i^{th} neighbor can be defined by:

$$d_i = \sqrt{\sum_{j=1}^{p}(x_{jf} - x_{ji})^2}$$

Prediction:

- Numerical y_f: take the average of the $y's$ in the k-nearest neighbors
- Categorical y_f: take the most common category in the k-nearest neighbors

k-Nearest Neighbors (kNN) – Example

Forensic Glass Analysis

Classifying shards of glass

Refractive index, plus oxide %

Na, Mg, Al, Si, K, Ca, Ba, Fe.

6 possible glass types

`WinF:` float glass window

`WinNF:` non-float window

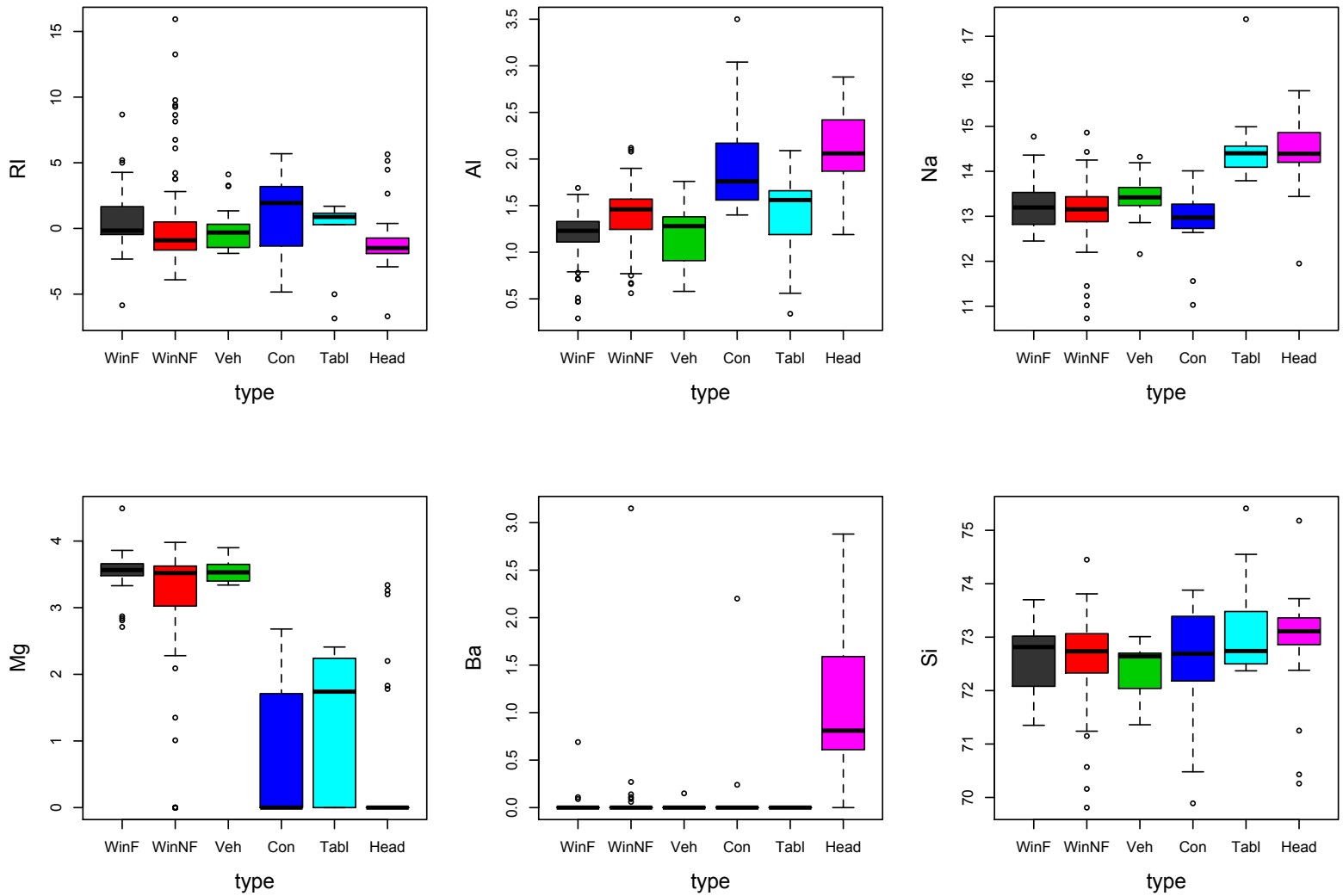
`Veh:` vehicle window

`Con:` container (bottles)

`Tabl:` tableware

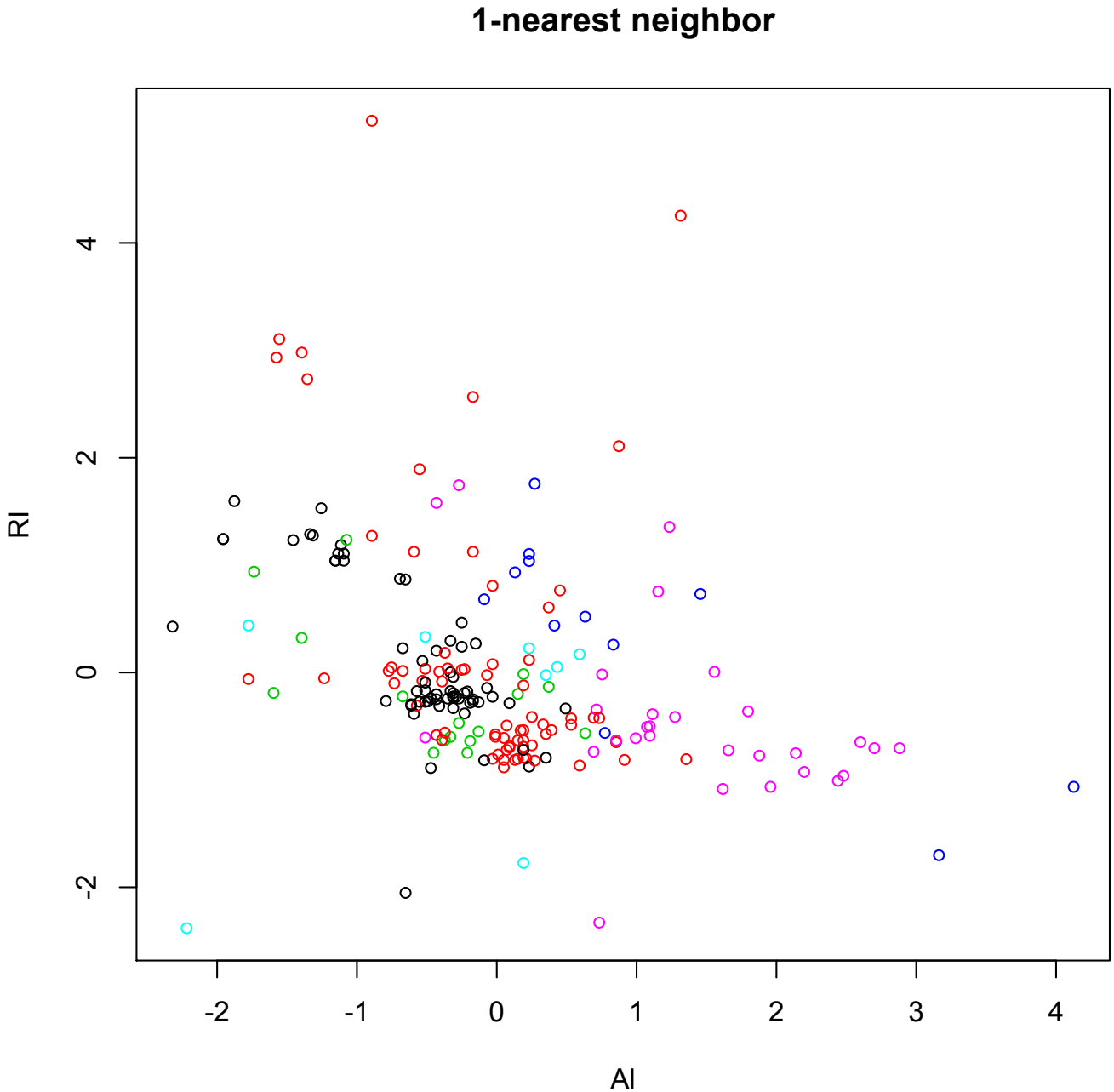
`Head:` vehicle headlamp

k-Nearest Neighbors (kNN) – Example

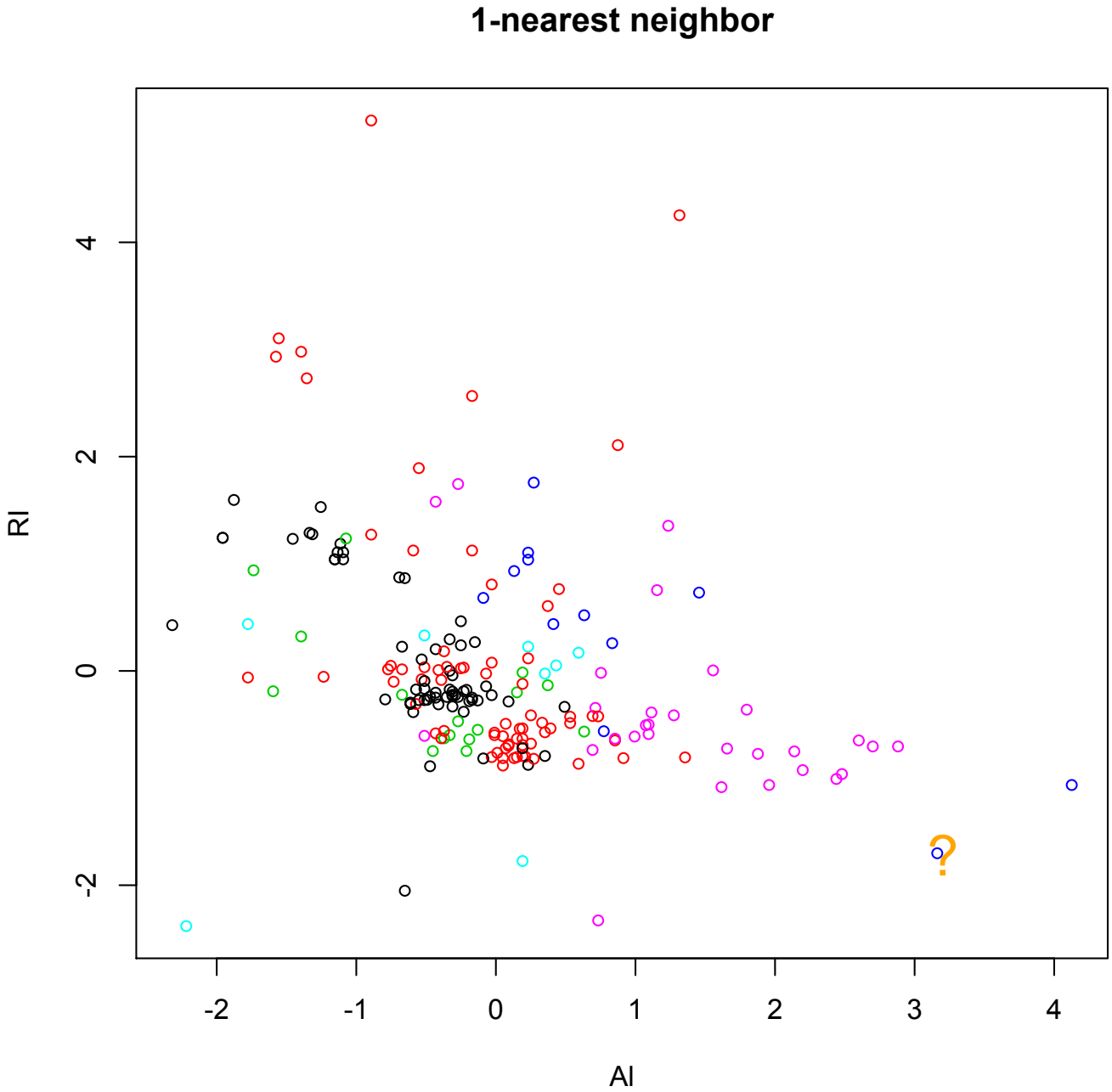


Some variables are clear discriminators (Ba) while others are more subtle (RI)

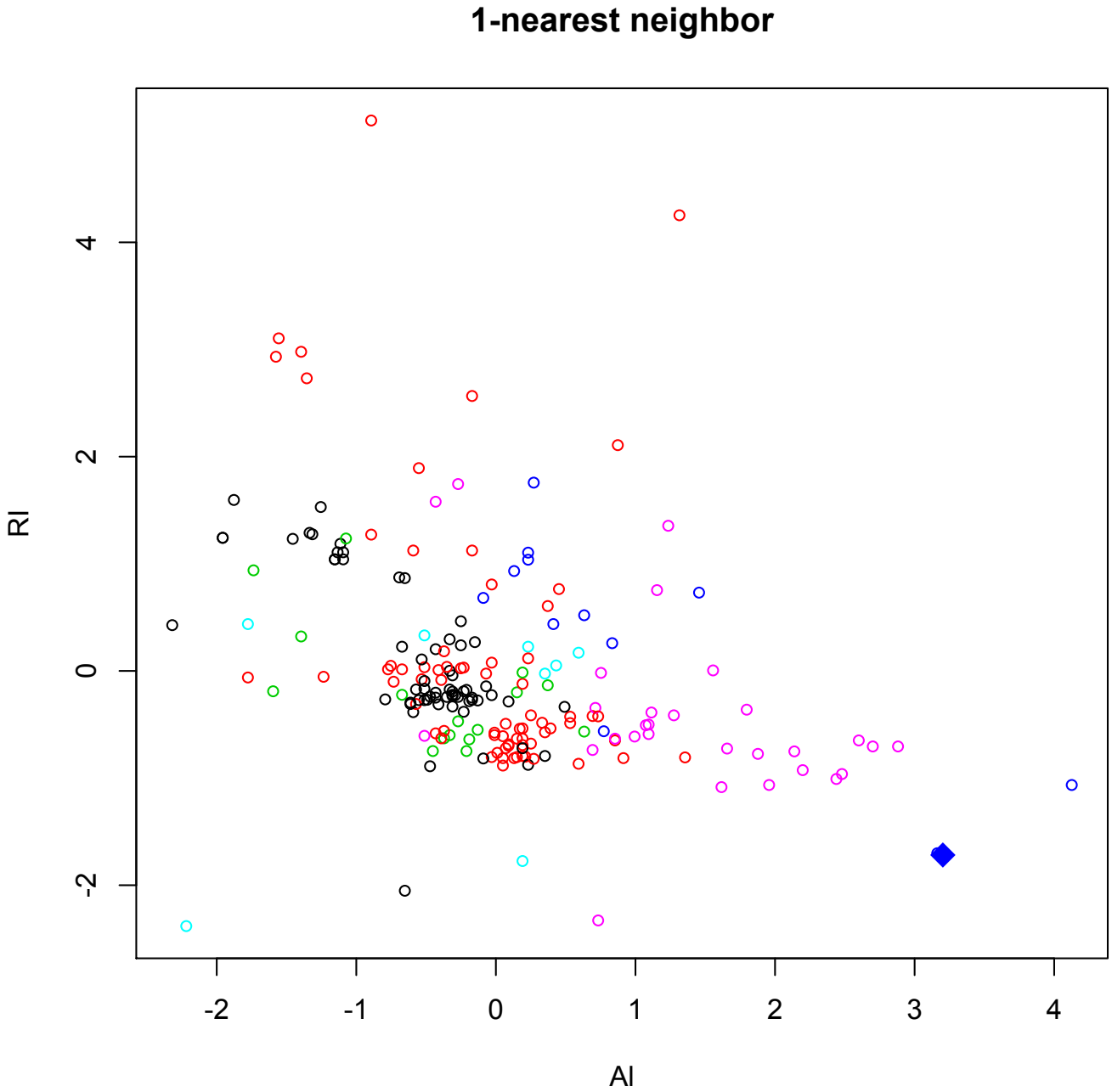
k-Nearest Neighbors (kNN) – Example



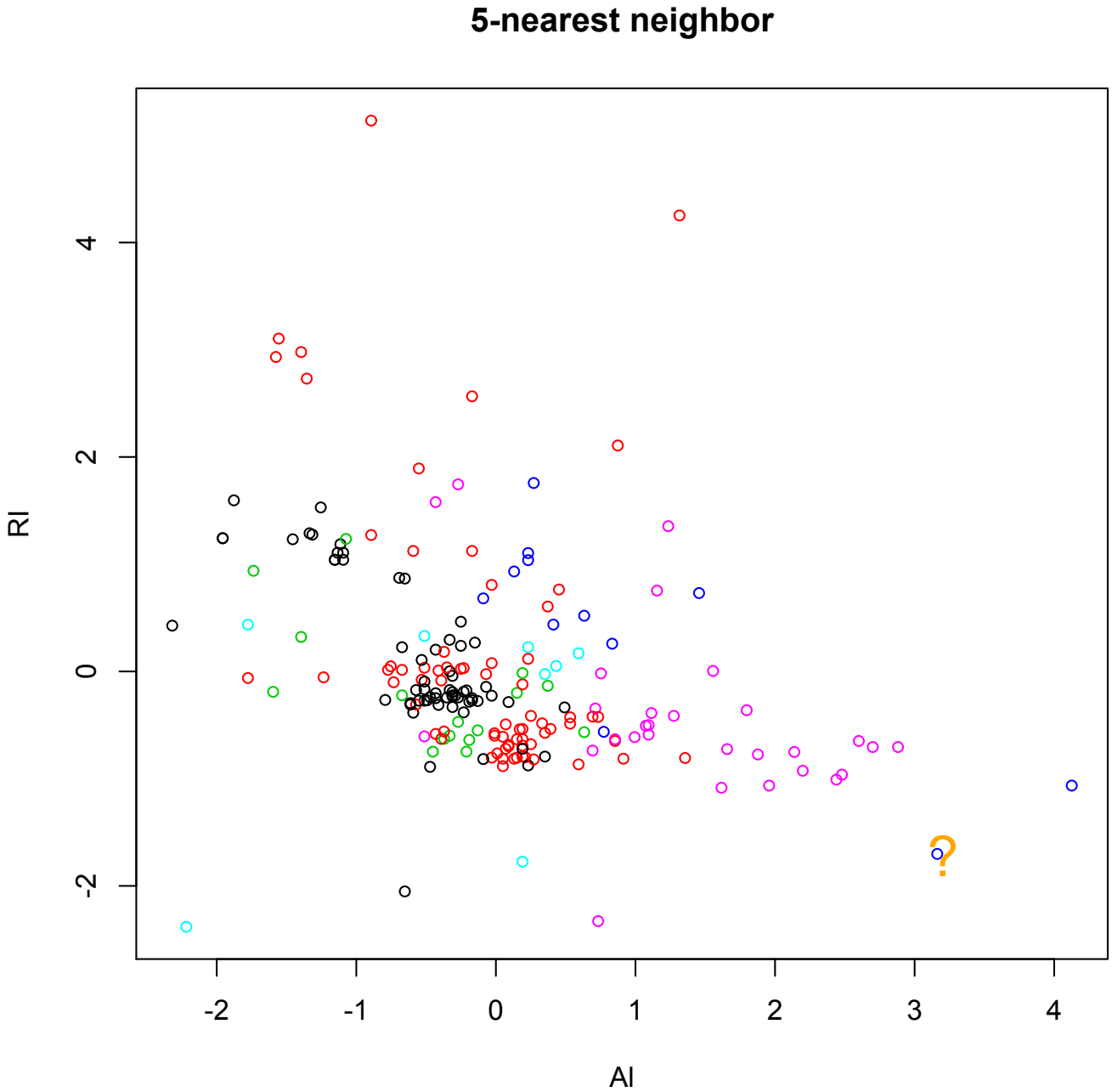
k-Nearest Neighbors (kNN) – Example



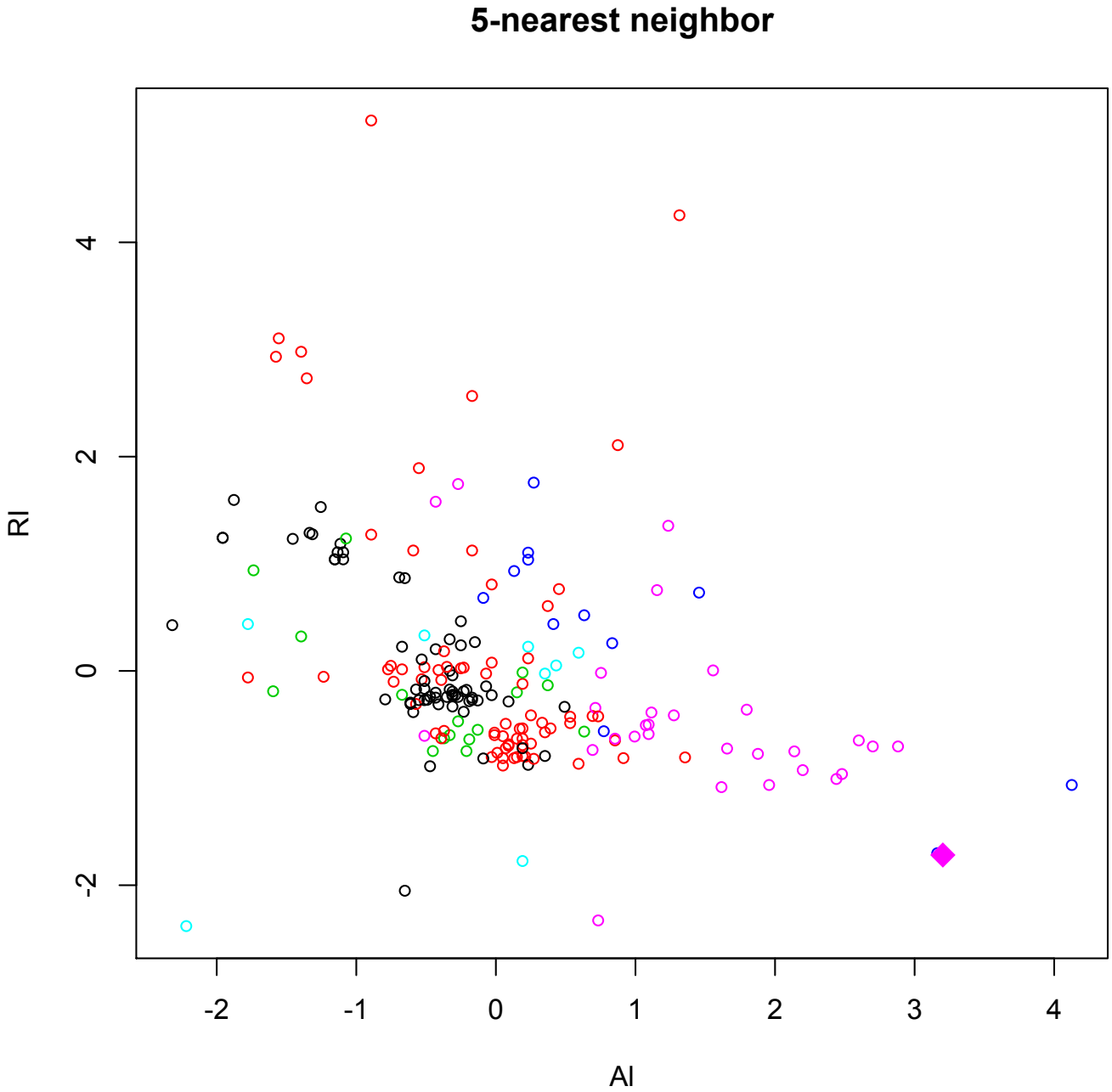
k-Nearest Neighbors (kNN) – Example



k-Nearest Neighbors (kNN) – Example



k-Nearest Neighbors (kNN) – Example



k-Nearest Neighbors (kNN)

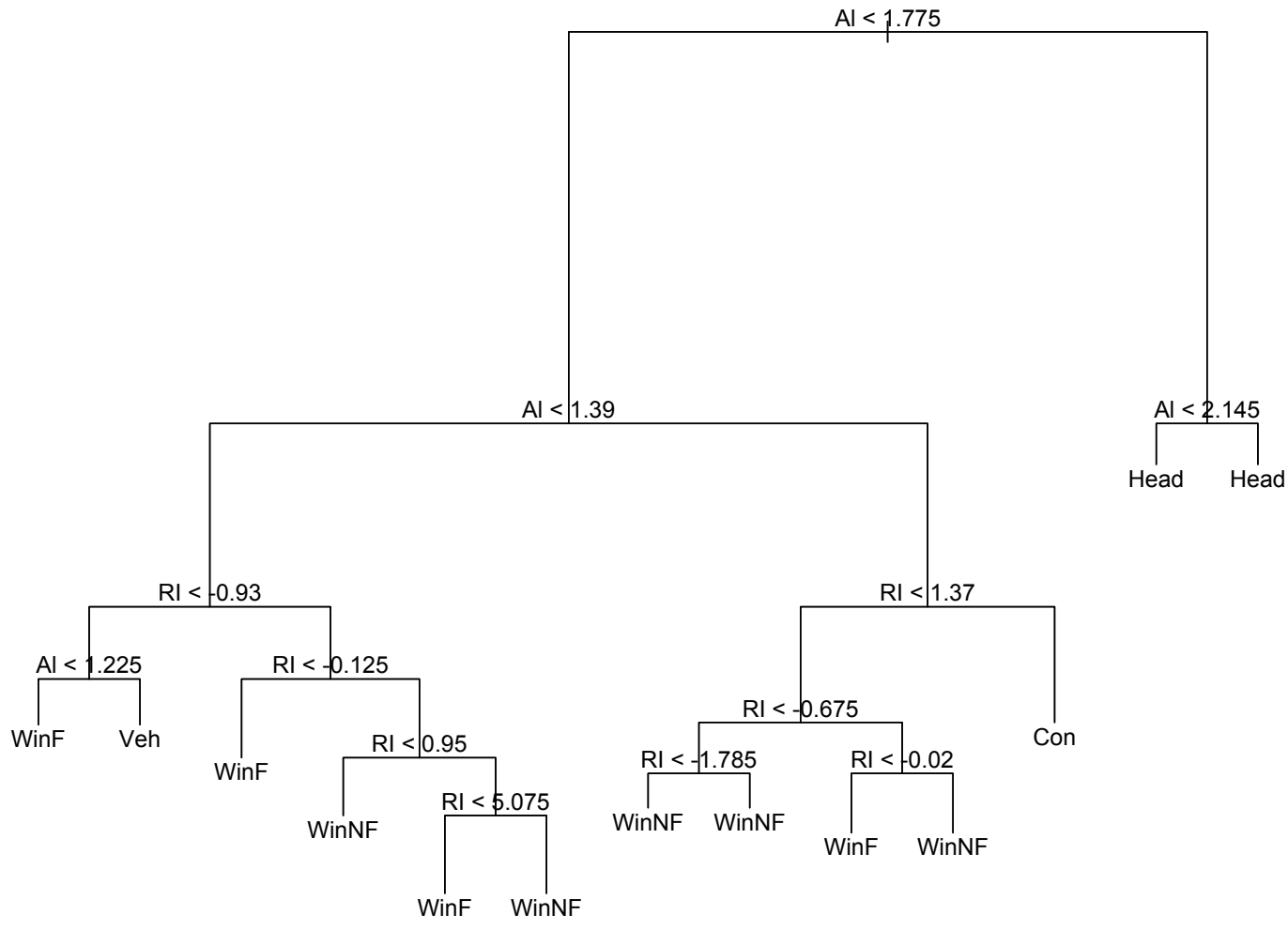
Comments:

- kNN is simple and intuitive and yet very powerful! e.g. Pandora, Nate Silver...
- Choice of k matters! Once again, we should rely on the out-of-sample performance
- Deciding the cut-off for classification impacts the results

- Can we contrast kNN to Logistic Regression?

Trees

Classification or Regression Trees are another example of flexible, interpretable and powerful tools for prediction.



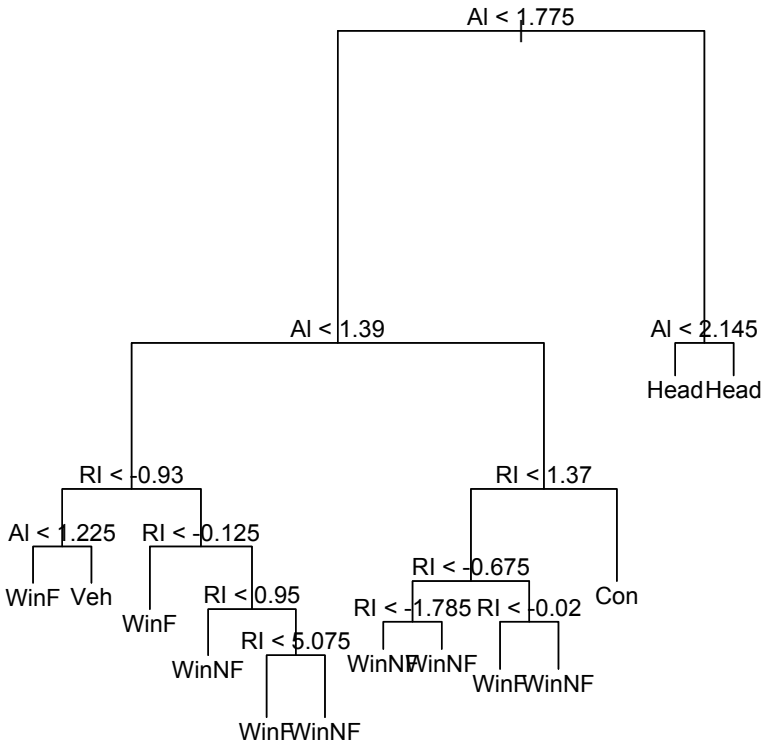
Trees

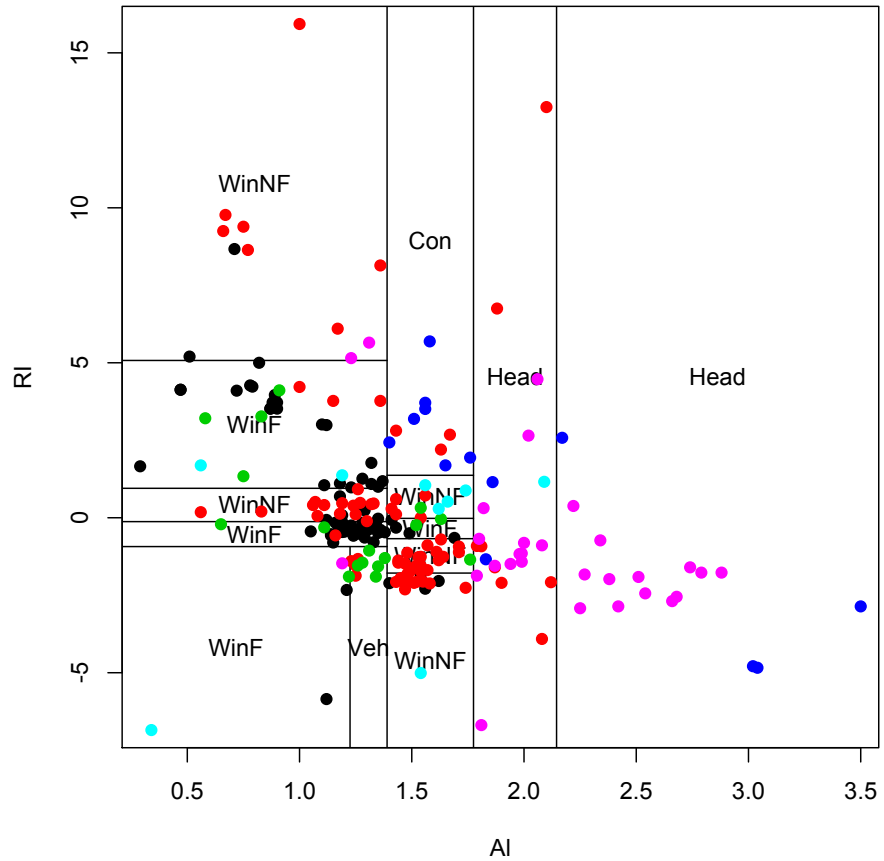
The main idea is to split the observations in the training data into subgroups by partitioning each predictor into subregions

These partitions create a sequence of logical rules that are intuitive, interpretable and easy to visualize!

Classification trees have class probabilities at the leaves.
Regression trees have a mean response at the leaves.

Trees – Glass Example





Growing Trees

Question: How do we choose a tree, ie, how do we define the partitions in the space of the covariates?

One very popular alternative is to use the **CART** algorithm. It is a recursive algorithm that goes as follows:

1. Choose the first split among all possible splits (for every x_i) that minimizes $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
   (or the error rate in classification trees)
2. For each newly defined subregion, repeat step 1.
3. Stop when no new split leads to a smaller sum of squared error (or error rate)

Tree Pruning: Improving prediction accuracy

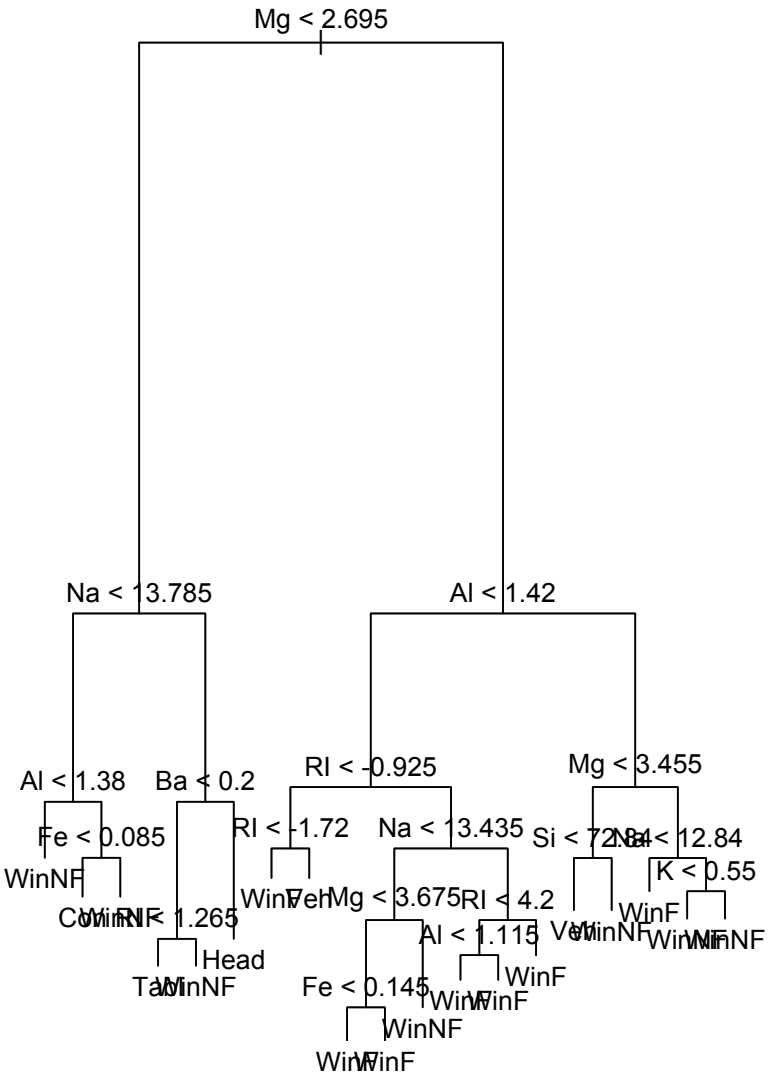
A large tree (too many terminal nodes) may over fit the training data! Usual problem with very flexible, large models...

Generally, one can improve the prediction ability of the model by "pruning" the tree, ie, cutting down some terminal nodes

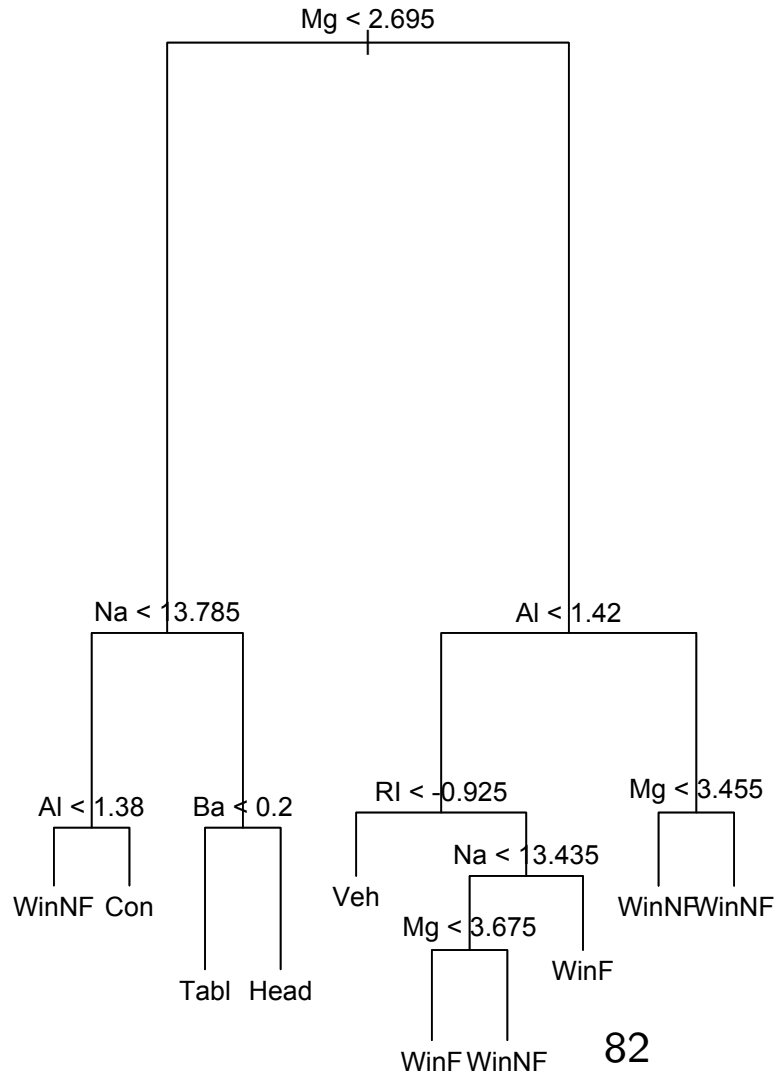
Again, this can (and should!) be done by comparing the out-of-sample prediction performance.

Tree Pruning: Improving prediction accuracy

Big tree Error Rate: 27%



Pruned Tree Error Rate: 20%



Trees – Comments

- Easy to explain and visualize
- Many modifications and options to "growing trees" are available. For example, we might want to control the minimum number of observations in each terminal node
- Lots of different algorithms available to grow trees
- Can be improved by mixing over a collection of trees... Random Forests, BART... the idea is that the combination of many simple trees might do better

California Housing

Data: Medium home values in census tract plus the following information:

- Location (latitude, longitude)
- Demographic information: population, income, etc...
- Average room/bedroom number, home age

Goal: Predict $log(MediumValue)$ (why logs?)

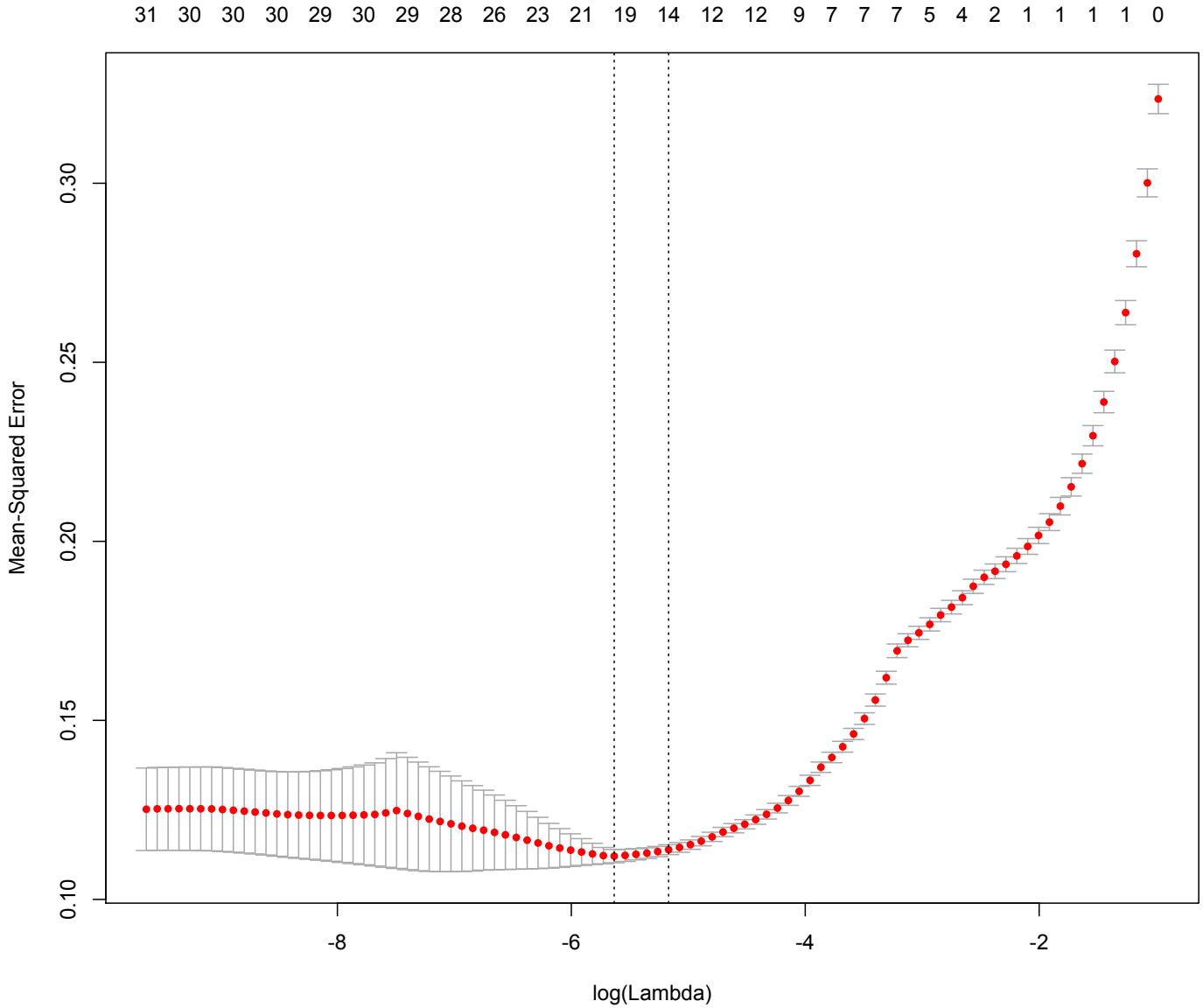
Would a linear model be appropriate here? Should the effect of each covariate be the same everywhere?

California Housing

Models: Let's compare the performance of the following models:

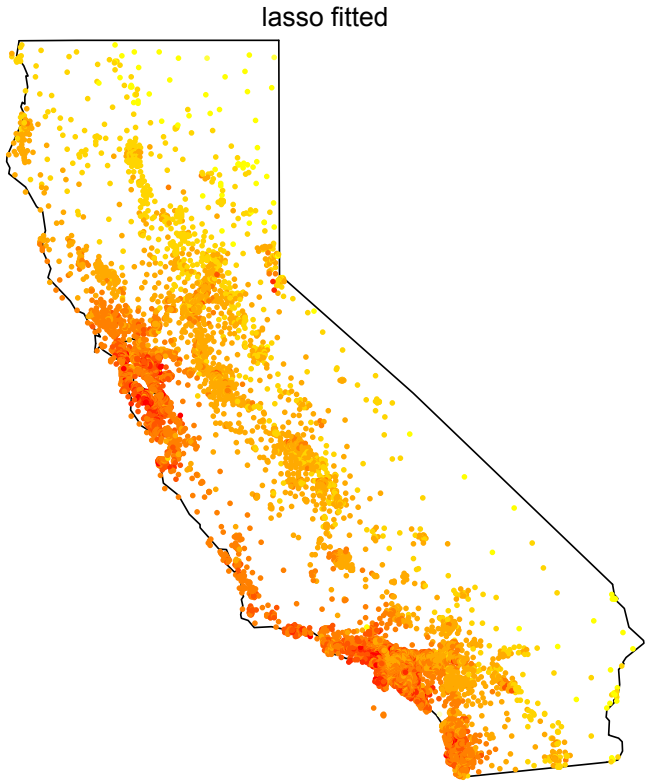
- Regression: LASSO plus interactions
- Regression Trees
- Random Forest

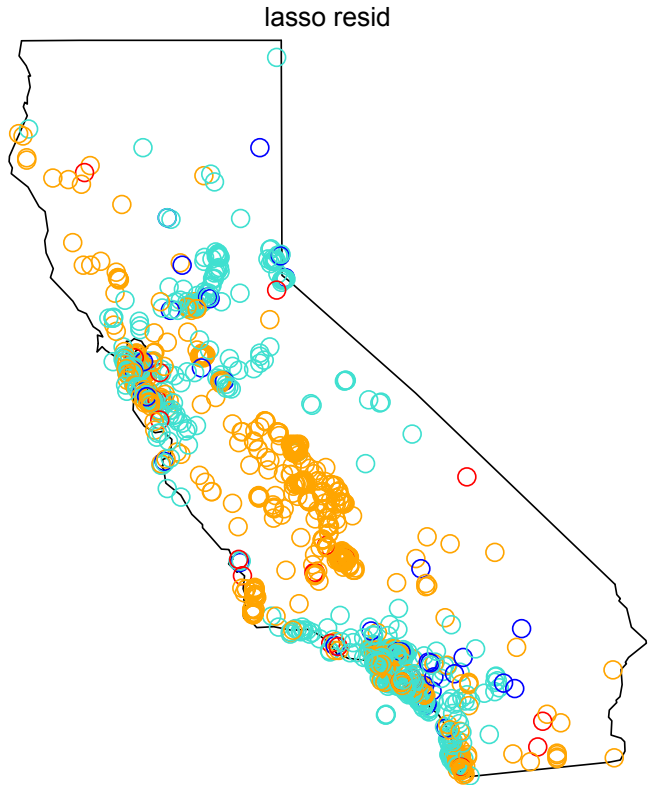
California Housing: LASSO



It chooses a very large model!

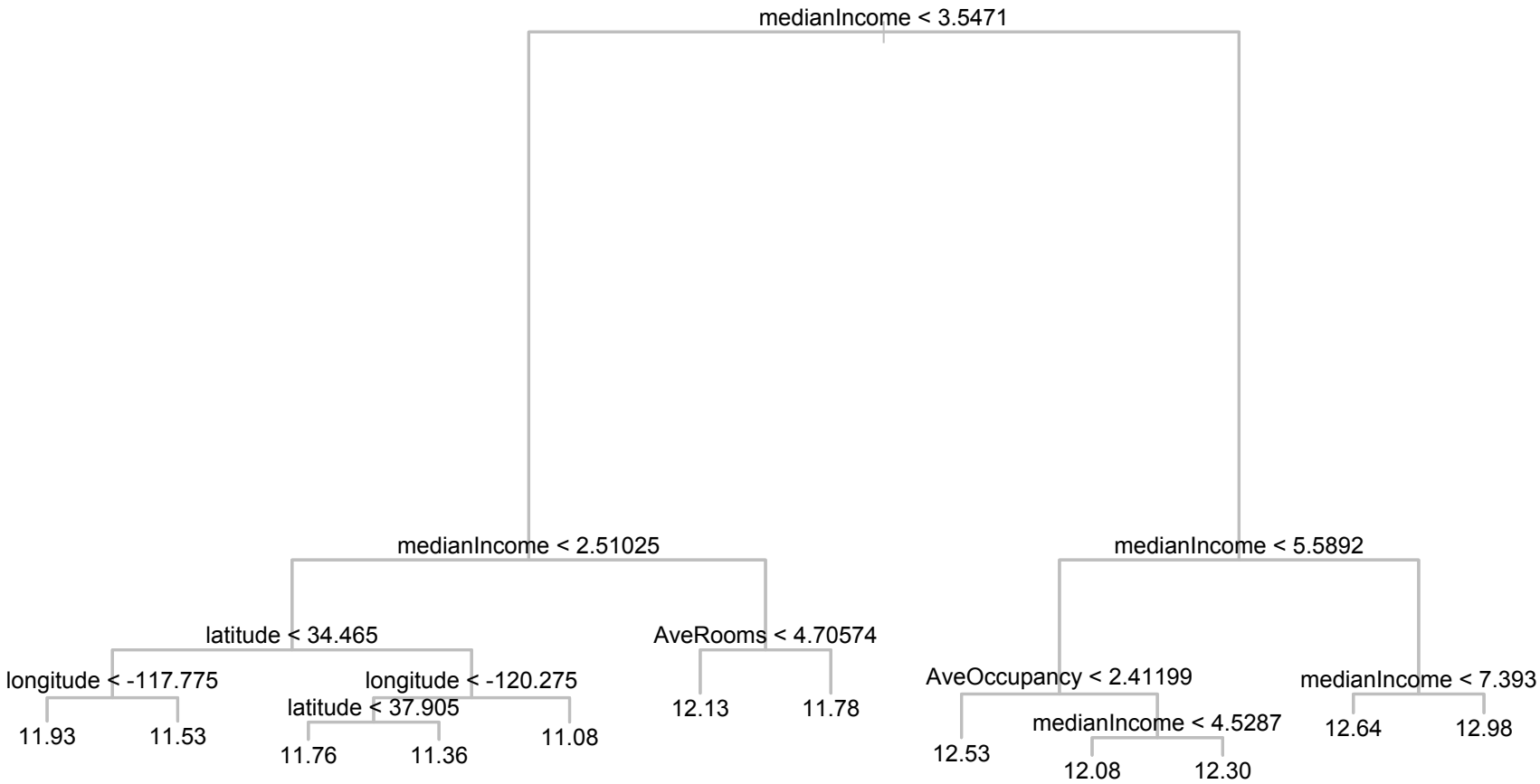
California Housing: LASSO



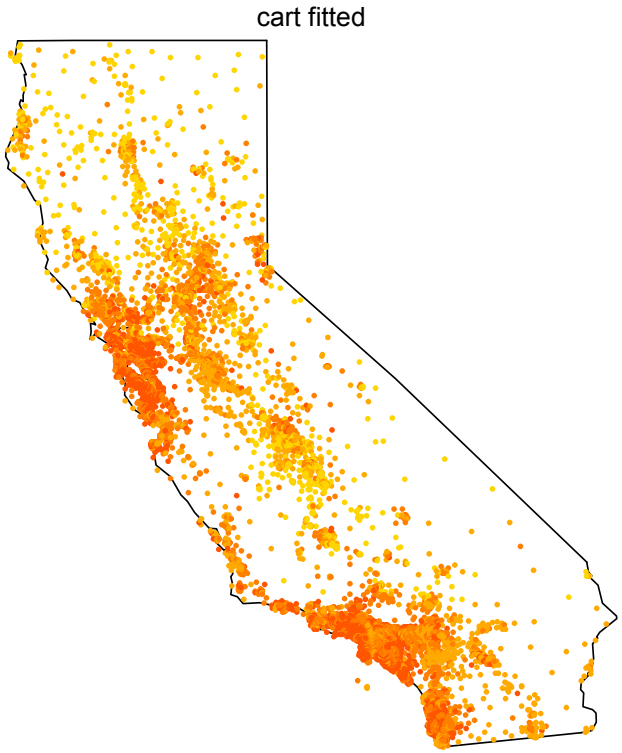


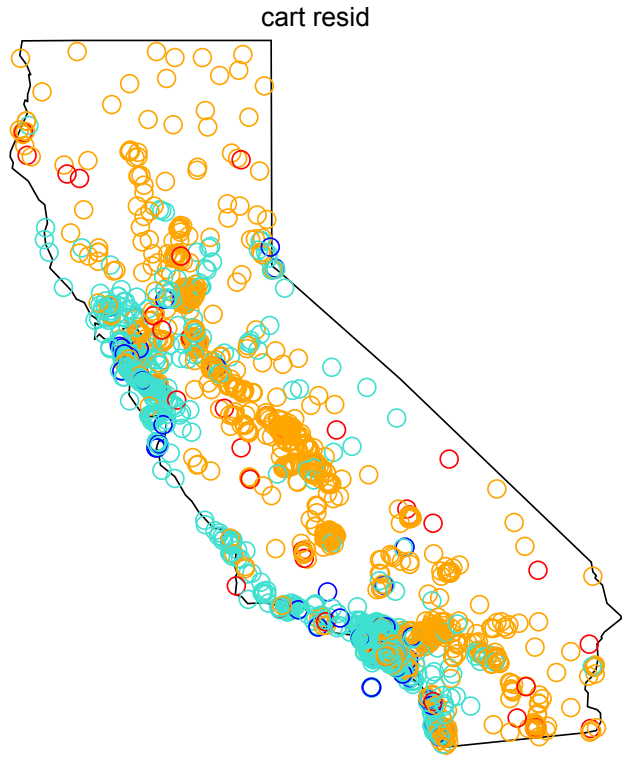
What do you see here... any patterns?

California Housing: Tree



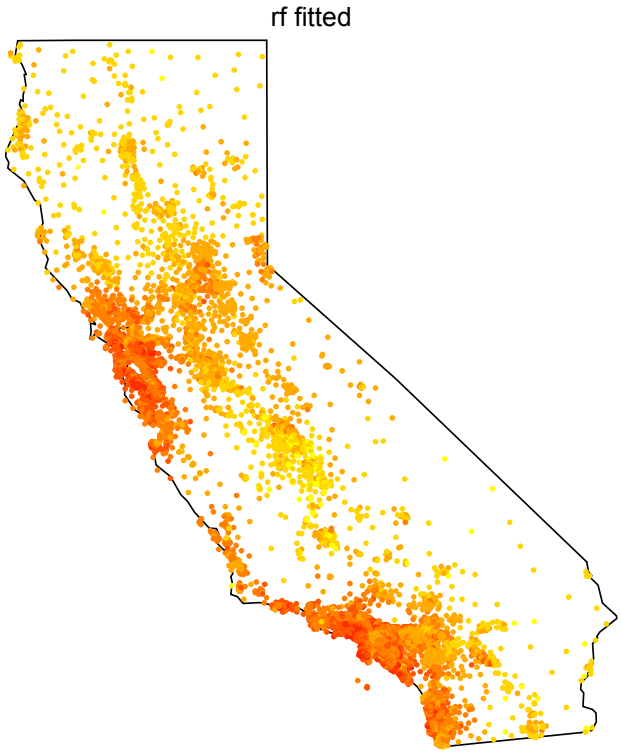
California Housing: Tree

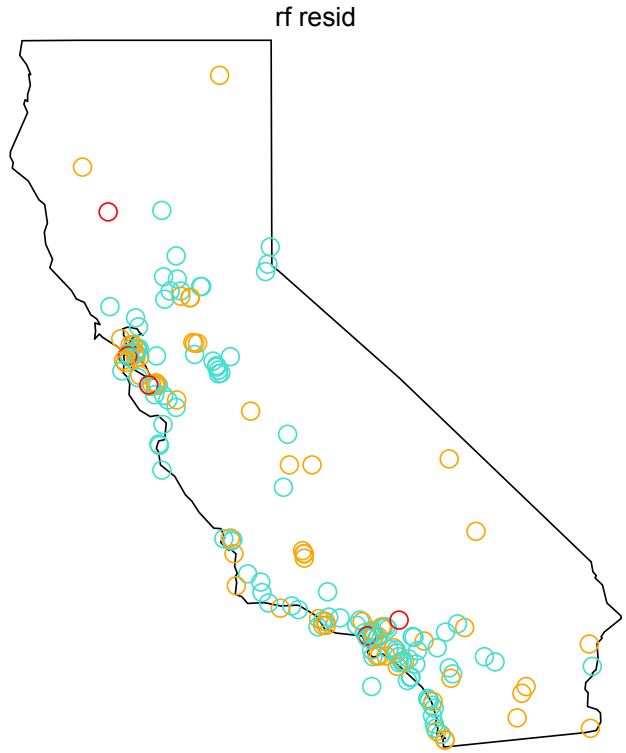




any better?

California Housing: Random Forest





Now?

California Housing: Out-of-Sample Performance