CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD
Cardiff Business School
Ysgol Busnes Caerdydd

Forecasting in R

Evaluating modeling accuracy

Bahman Rostami-Tabar

Outline

1. Residual diagnostics
2. Evaluating point forecast accuracy
3. Time series cross validation
4. Evaluating prediction interval accuracy
5. Lab session 6

Outline

1. Residual diagnostics
2. Evaluating point forecast accuracy
3. Time series cross validation
4. Evaluating prediction interval accuracy
5. Lab session 6

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

Assumptions

1. $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
2. $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Forecasting residuals

Residuals in forecasting: difference between observed value and its fitted value: $e_t = y_t - \hat{y}_{t|t-1}$.

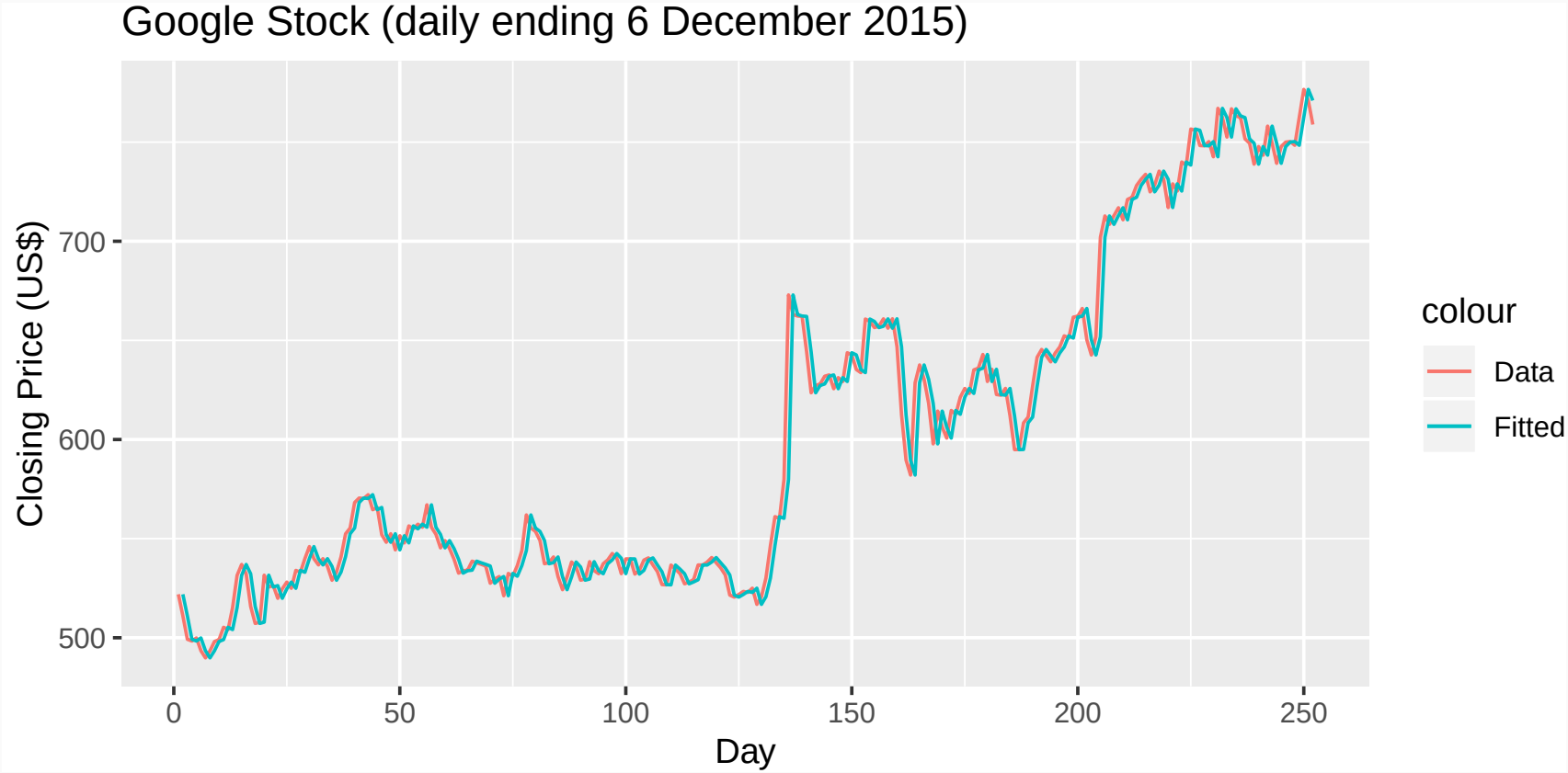
Assumptions

1. $\{e_t\}$ uncorrelated. If they aren't, then information left in residuals that should be used in computing forecasts.
2. $\{e_t\}$ have mean zero. If they don't, then forecasts are biased.

Useful properties (for prediction intervals)

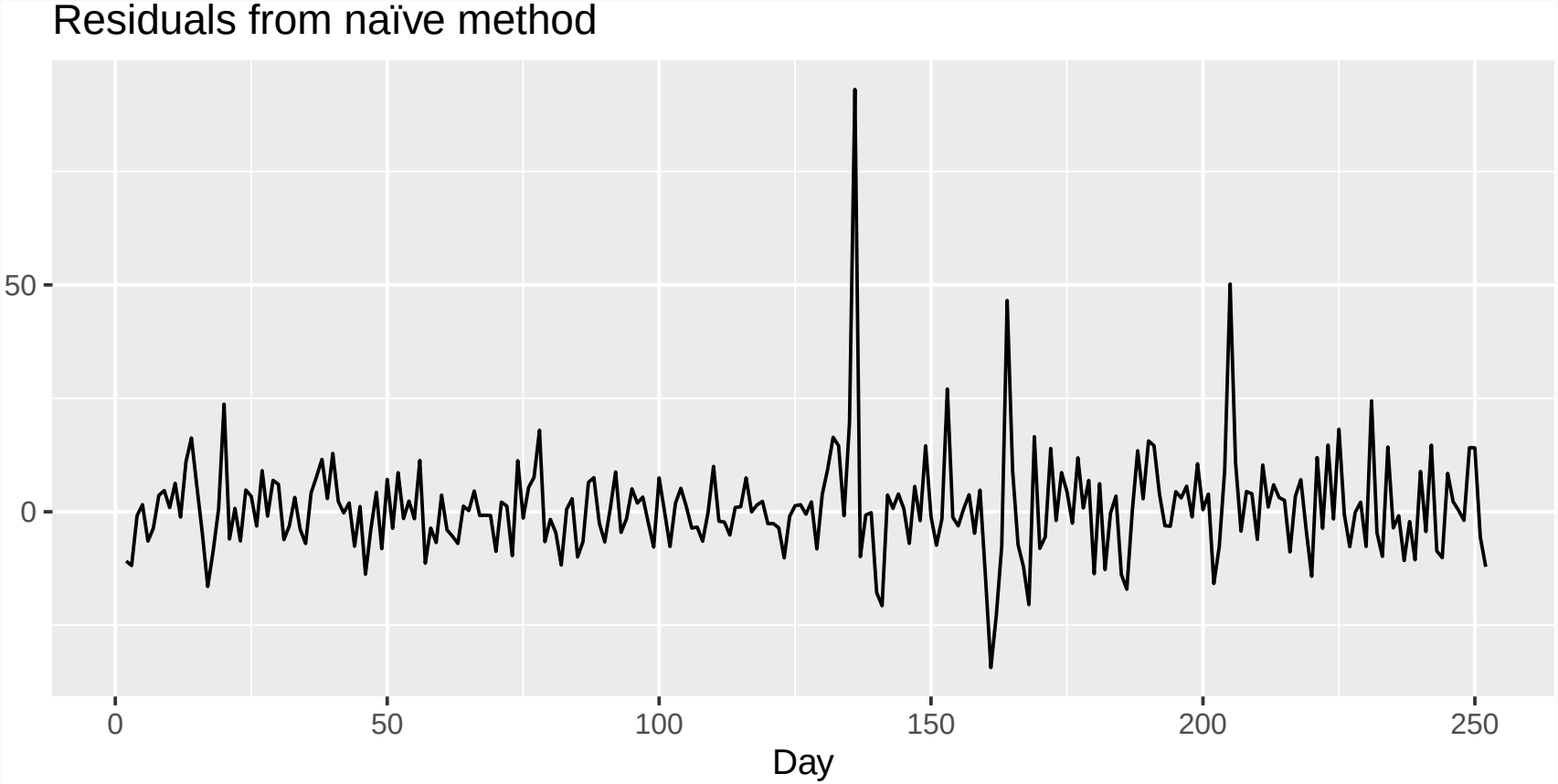
3. $\{e_t\}$ have constant variance.
4. $\{e_t\}$ are normally distributed.

Example: Google stock price



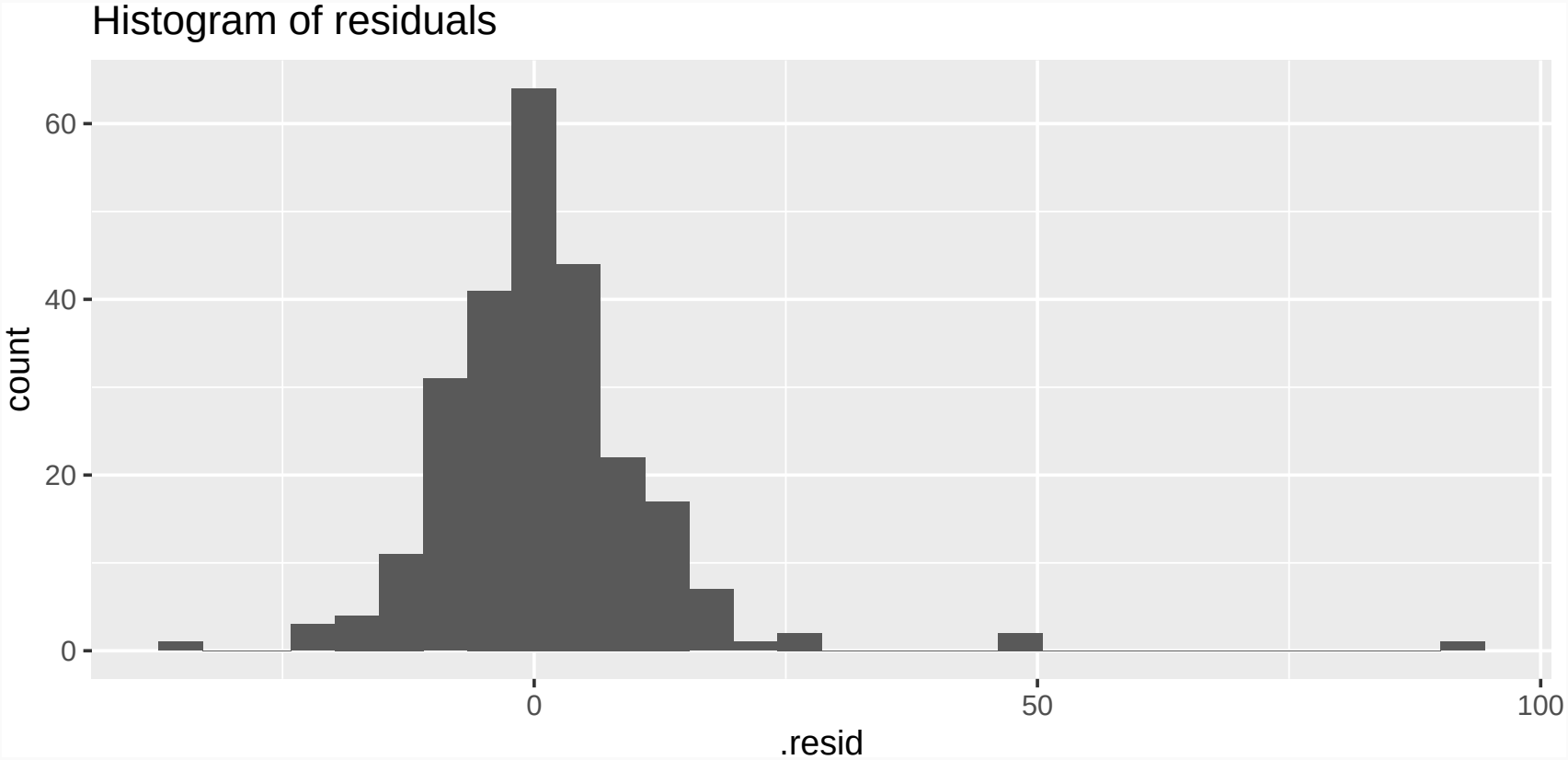
Example: Google stock price

```
augment(fit) %>%
  autoplot(.resid) + xlab("Day") + ylab("") +
    ggtitle("Residuals from naïve method")
```



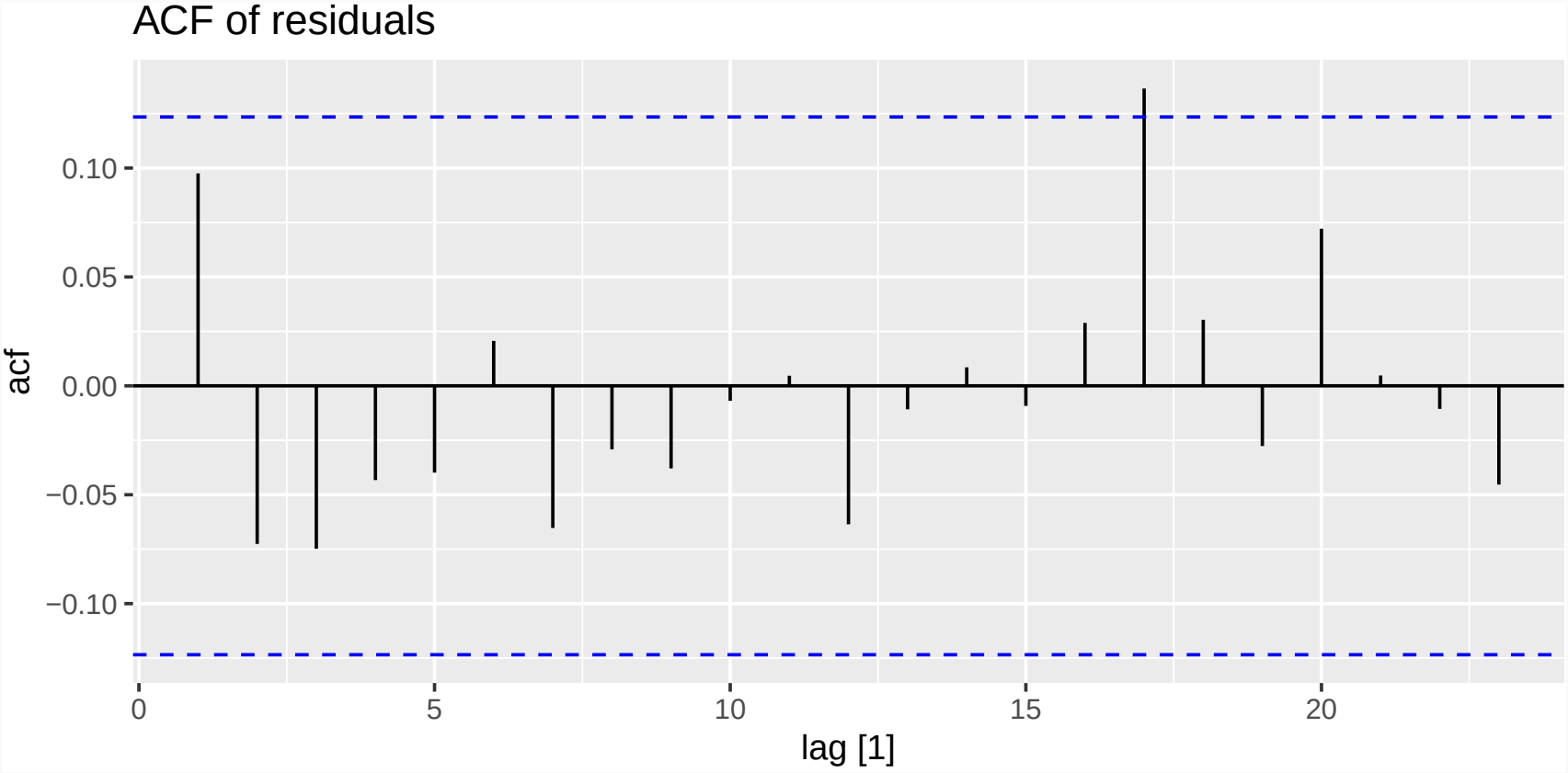
Example: Google stock price

```
augment(fit) %>%
  ggplot(aes(x = .resid)) +
    geom_histogram(bins = 30) +
    ggtitle("Histogram of residuals")
```



Example: Google stock price

```
augment(fit) %>% ACF(.resid) %>%
  autoplot() + ggtitle("ACF of residuals")
```



ACF of residuals

- We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren't, then there is information left in the residuals that should be used in computing forecasts.
- So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.
- We *expect* these to look like white noise.

Portmanteau tests

Consider a *whole set* of r_k values, and develop a test to see whether the set is significantly different from a zero set.

Portmanteau tests

Consider a *whole set* of r_k values, and develop a test to see whether the set is significantly different from a zero set.

Box-Pierce test

$$Q = T\sum_{k=1}^{h} r_k^2$$

where h is max lag being considered and T is number of observations.

- If each r_k close to zero, Q will be **small**.
- If some r_k values large (positive or negative), Q will be **large**.

Portmanteau tests

Consider a *whole set* of r_k values, and develop a test to see whether the set is significantly different from a zero set.

Ljung-Box test

$$Q^* = T(T+2)\sum_{k=1}^{h}(T-k)^{-1}r_k^2$$

where h is max lag being considered and T is number of observations.

- Preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- Better performance, especially in small samples.

Portmanteau tests

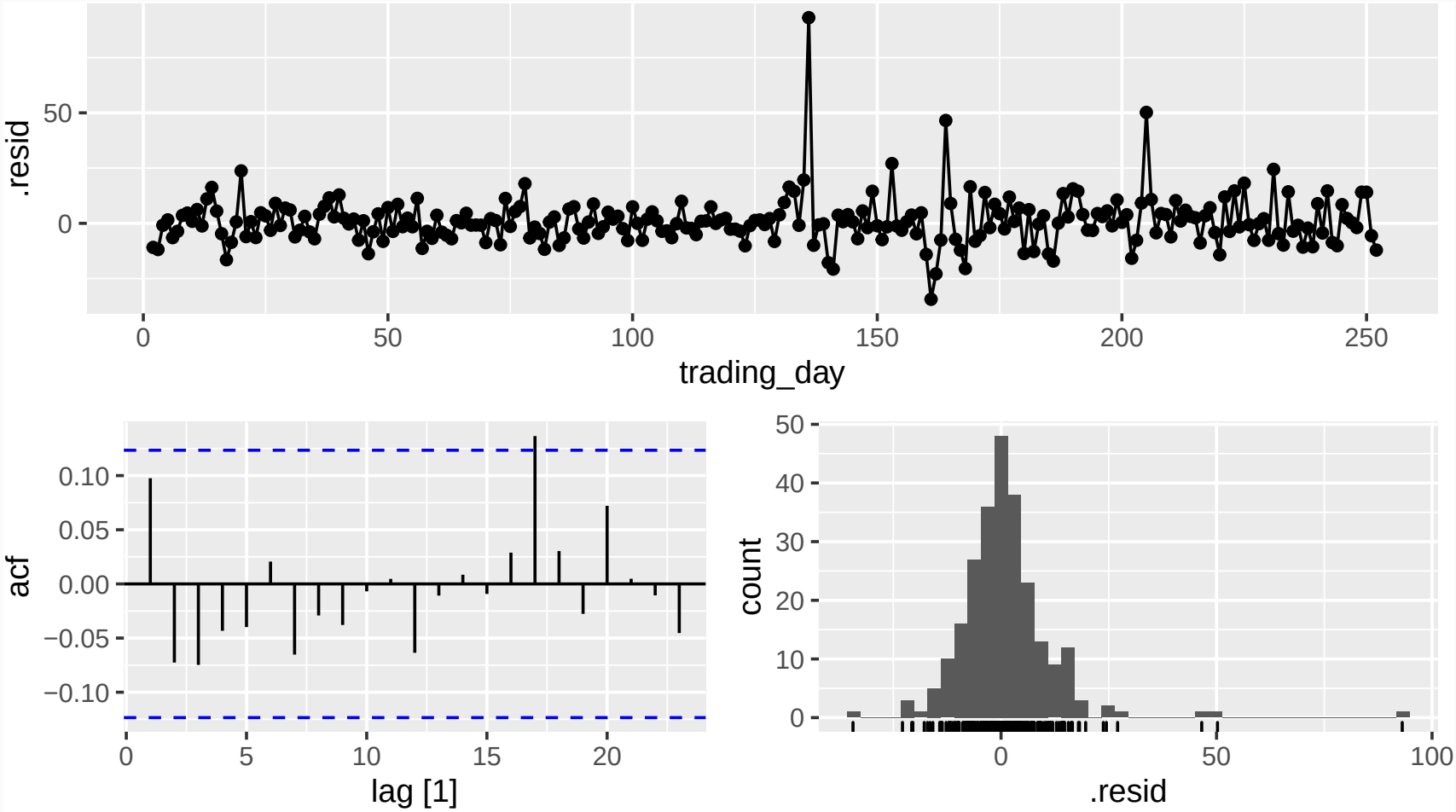
- If data are WN, Q^* has χ^2 distribution with $(h - K)$ degrees of freedom where K = no. parameters in model.
- When applied to raw data, set $K = 0$.

```
augment(fit) %>% features(.resid, ljung_box, lag=10,dof=0)
```

```
## # A tibble: 1 x 4
##   Symbol .model          lb_stat lb_pvalue
##   <chr>  <chr>             <dbl>     <dbl>
## 1 GOOG   NAIVE(Close)      7.91     0.637
```

gg_tsresiduals function

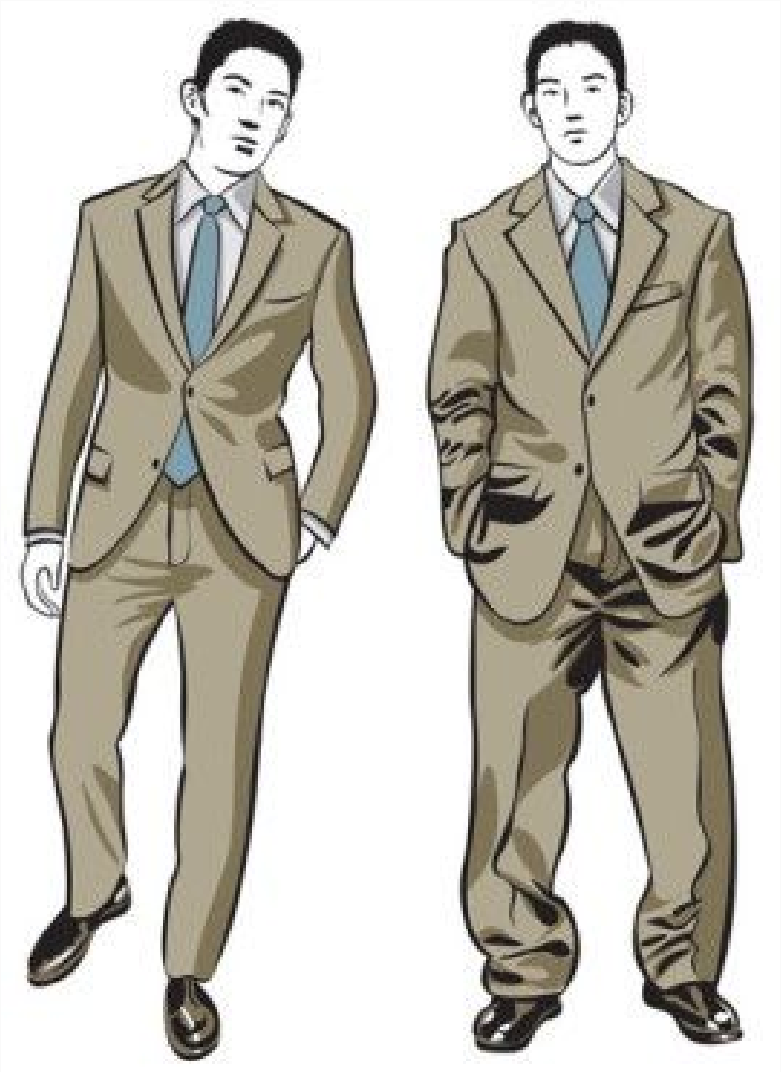
```
fit %>%
  gg_tsresiduals()
```



Outline

1. Residual diagnostics
2. Evaluating point forecast accuracy
3. Time series cross validation
4. Evaluating prediction interval accuracy
5. Lab session 6

Good fit versus forecast accuracy



Evaluate forecast accuracy

- Residual diagnostic is not a reliable indication of forecast accuracy
- A model which fits the training data well will not necessarily forecast well
- A perfect fit can always be obtained by using a model with enough parameters
- Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data

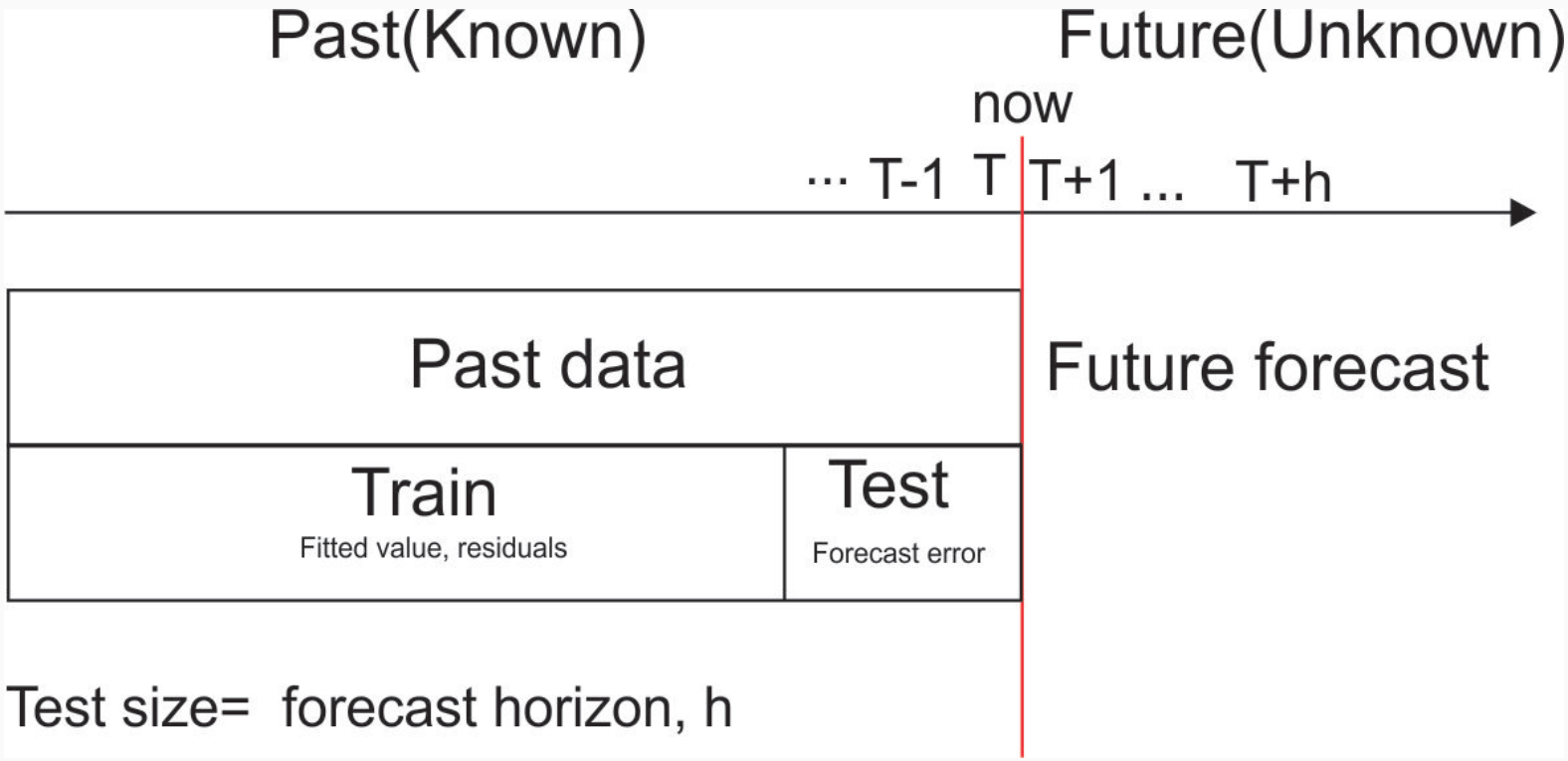
Evaluate forecast accuracy

The accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when fitting the model

Training and test sets

- We mimic the real life situation
- We pretend we don’t know some part of data(new data)
- It must not be used for *any* aspect of model training
- Forecast accuracy is based only on the test set

Training and test series



Split the data

use fucntions in `dplyr` and `lubridate` such as `filter`, `filter_index`, `slice`, `year`

```
# Filter the year of interest
google_2015 <- google_stock %>%
  filter_index("2015")
```

Forecast errors

Forecast "error": the difference between an observed value and its forecast

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

where the training data is given by $\{y_1, \ldots, y_T\}$

- Unlike residuals, forecast errors on the test set involve multi-step forecasts.
- These are *true* forecast errors as the test data is not used in computing $\hat{y}_{T+h|T}$.

Measures of forecast accuracy

Measures of forecast accuracy

y_{T+h} = $(T+h)$th observation, $h = 1, \ldots, H$

$\hat{y}_{T+h|T}$ = its forecast based on data up to time T.

e_{T+h} = $y_{T+h} - \hat{y}_{T+h|T}$

$$\text{MAE} = \text{mean}(|e_{T+h}|)$$

$$\text{MSE} = \text{mean}(e_{T+h}^2) \qquad \text{RMSE} = \sqrt{\text{mean}(e_{T+h}^2)}$$

$$\text{MAPE} = 100\text{mean}(|e_{T+h}|/|y_{T+h}|)$$

Measures of forecast accuracy

y_{T+h} = $(T+h)$th observation, $h = 1, \ldots, H$

$\hat{y}_{T+h|T}$ = its forecast based on data up to time T.

e_{T+h} = $y_{T+h} - \hat{y}_{T+h|T}$

$$\text{MAE} = \text{mean}(|e_{T+h}|)$$

$$\text{MSE} = \text{mean}(e_{T+h}^2) \qquad \text{RMSE} = \sqrt{\text{mean}(e_{T+h}^2)}$$

$$\text{MAPE} = 100\text{mean}(|e_{T+h}|/|y_{T+h}|)$$

- MAE, MSE, RMSE are all scale dependent
- MAPE is scale independent but is only sensible if $y_t \gg 0$ for all t, and y has a natural zero.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}|/Q)$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

For non-seasonal time series,

$$Q = (T-1)^{-1} \sum_{t=2}^{T} |y_t - y_{t-1}|$$

works well. Then MASE is equivalent to MAE relative to a naïve method.

Measures of forecast accuracy

Mean Absolute Scaled Error

$$\text{MASE} = \text{mean}(|e_{T+h}|/Q)$$

where Q is a stable measure of the scale of the time series $\{y_t\}$.

For seasonal time series,

$$Q = (T-m)^{-1} \sum_{t=m+1}^{T} |y_t - y_{t-m}|$$

works well. Then MASE is equivalent to MAE relative to a seasonal naïve method.

Test set accuracy

```
accuracy(google_fc, google_stock)
```

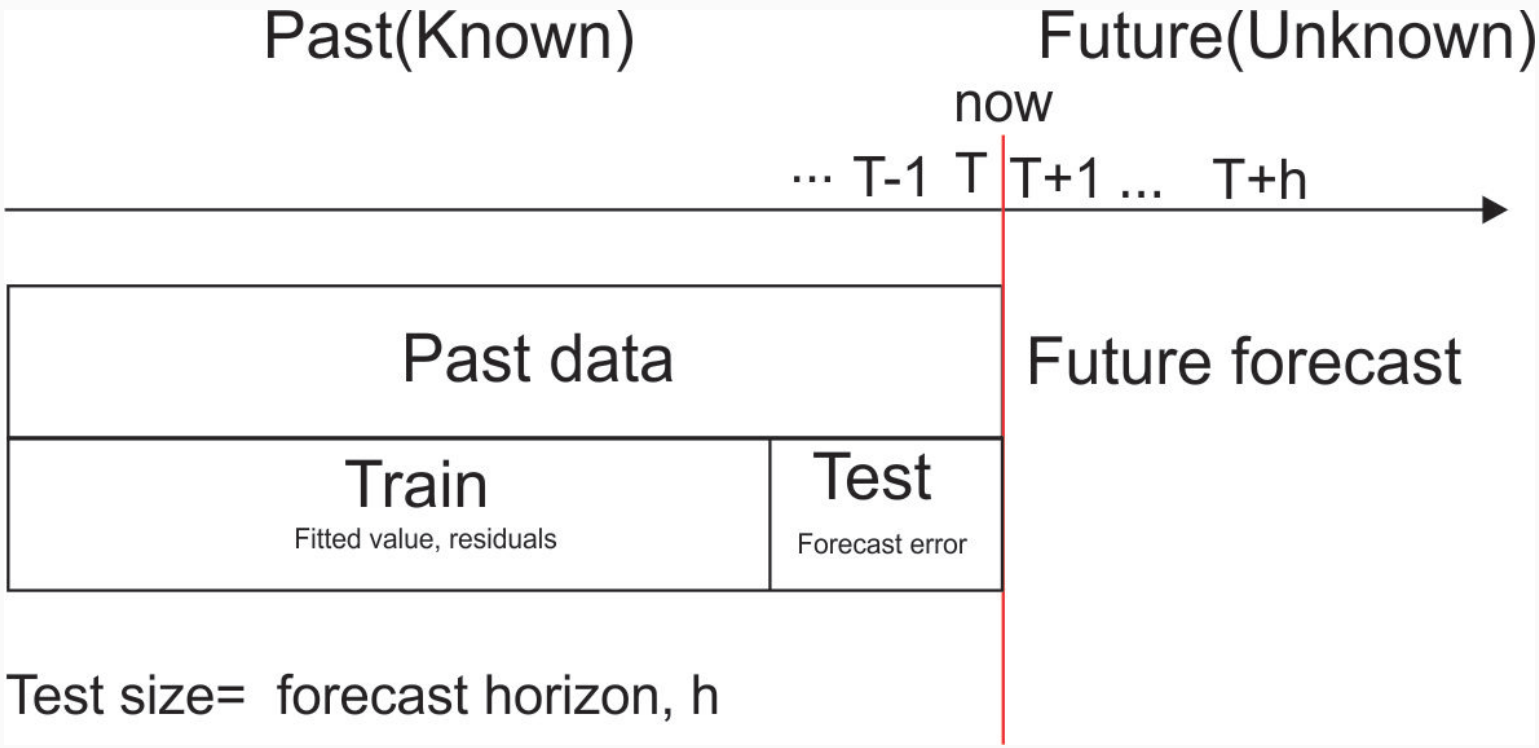
	RMSE	MAE	MASE
drift method	53.06958	49.82414	7.841618
mean method	118.03221	116.94525	18.405533
naive method	43.43152	40.38421	6.355906

Poll: true or false?

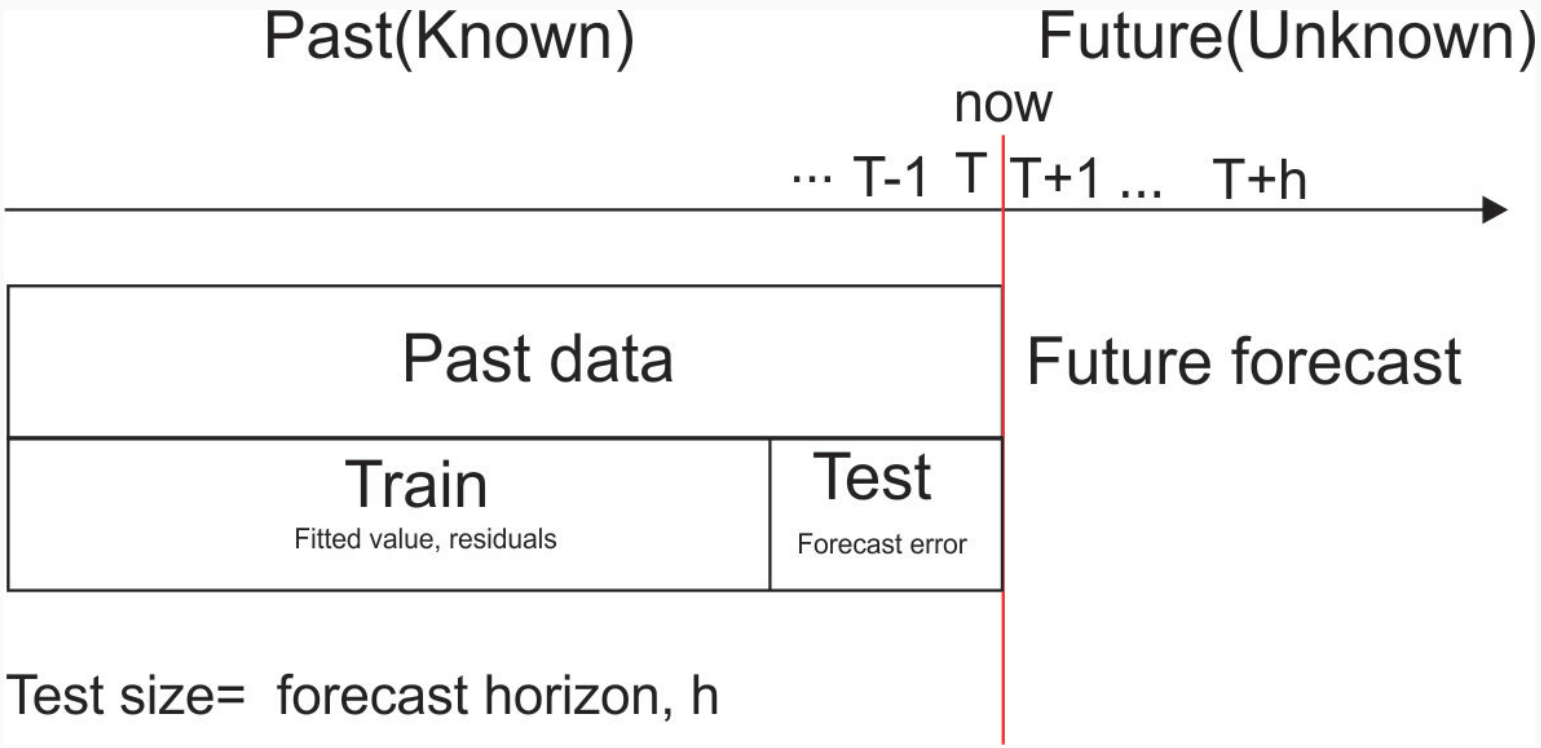
1. Good point forecast models should have normally distributed residuals.
2. A model with small residuals will give good forecasts.
3. The best measure of forecast accuracy is MAPE.
4. Always choose the model with the best forecast accuracy as measured on the test set.

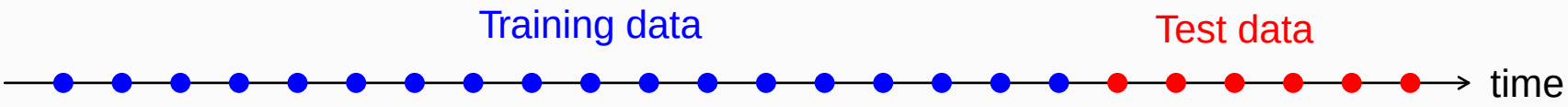
Outline

Issue with traditional evaluation

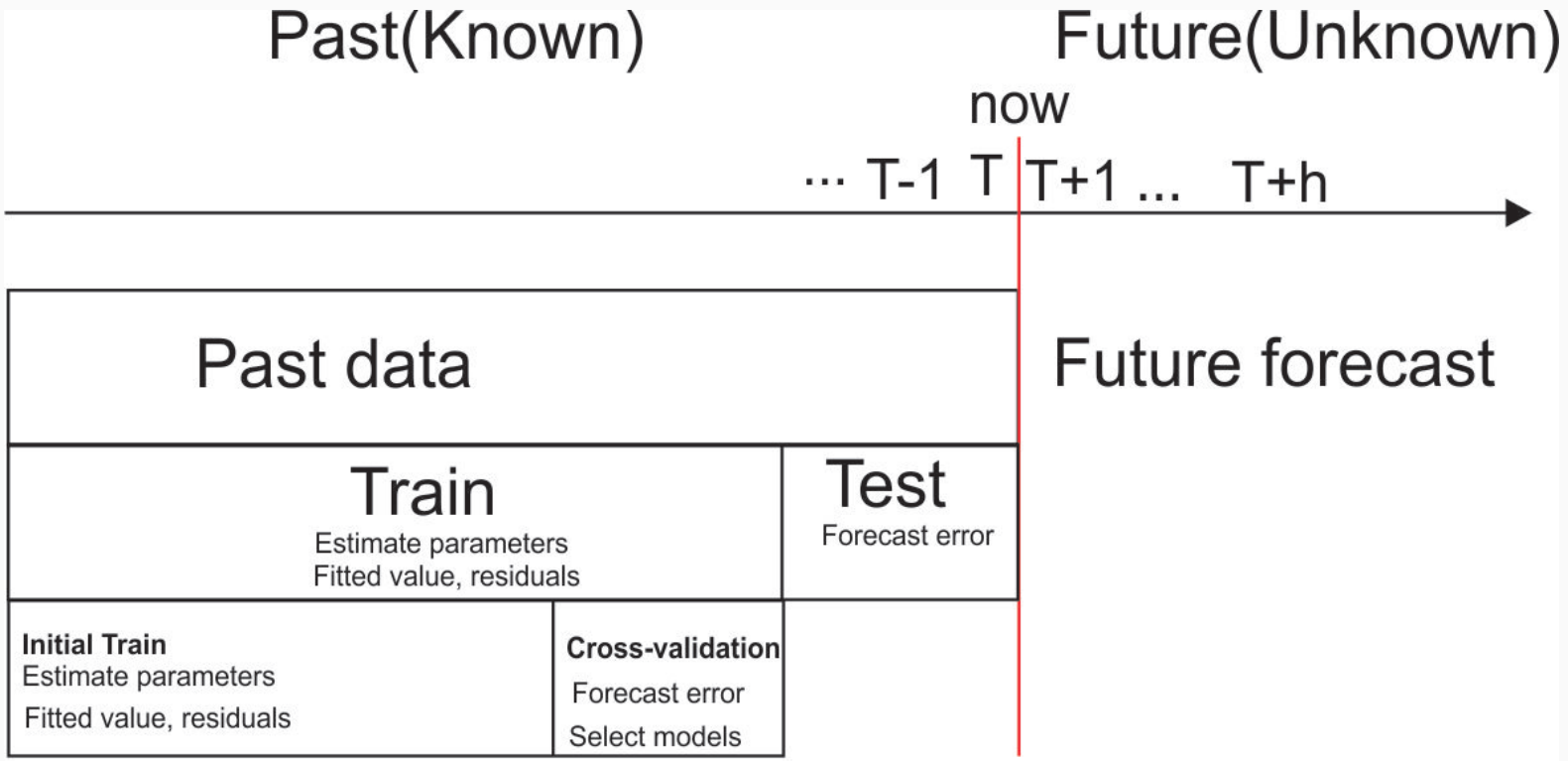


Issue with traditional evaluation





Time series cross-validation

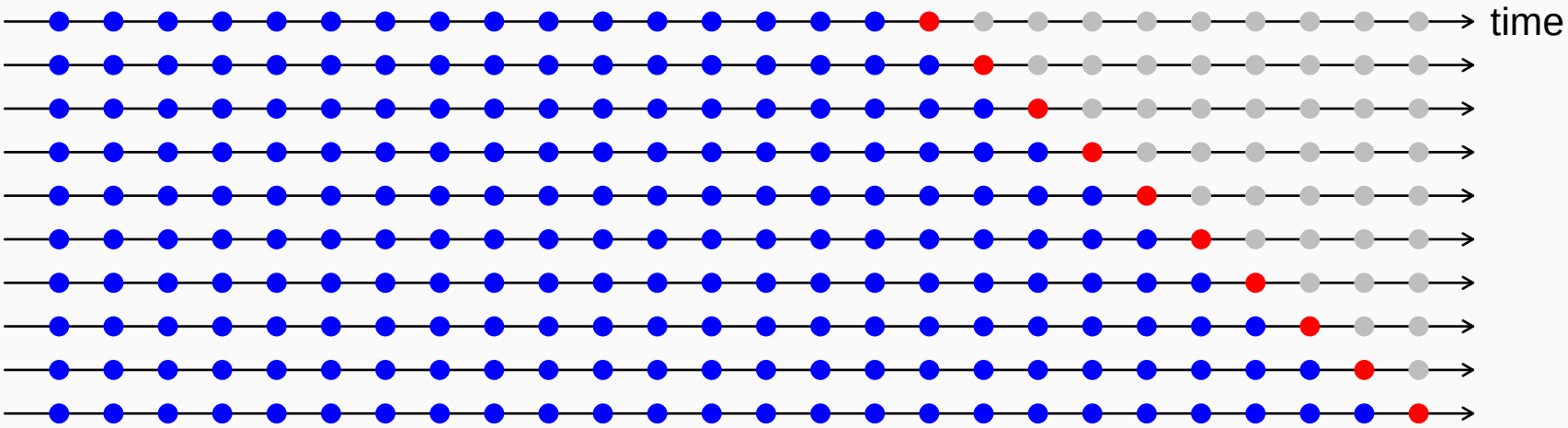


Test size= forecast horizon, h

Cross-validation size=nb of experiment+h-1

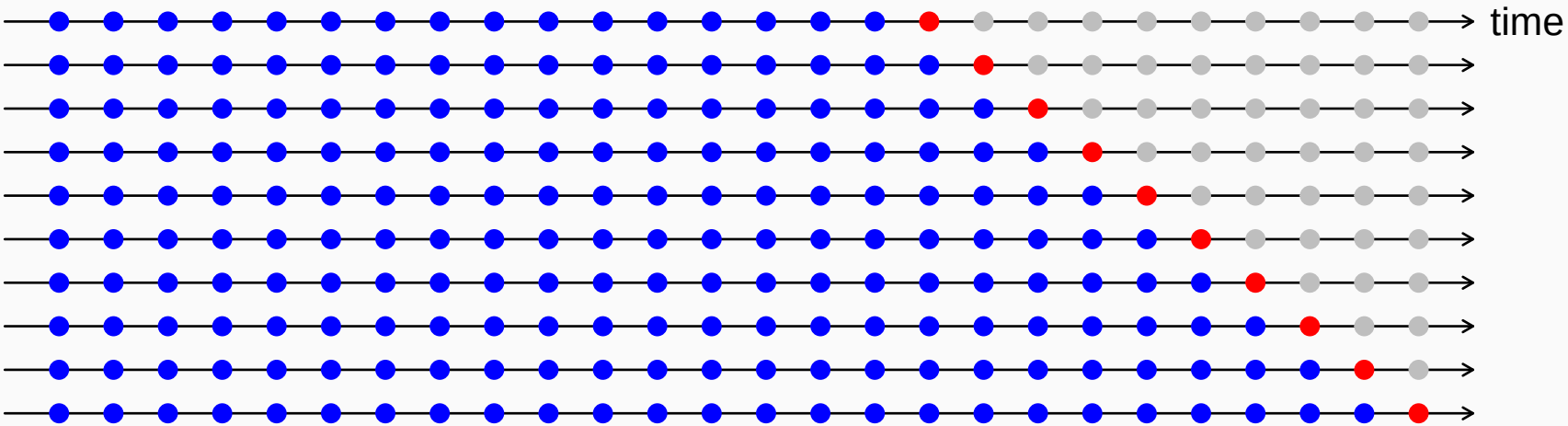
Time series cross-validation

Time series cross-validation



Time series cross-validation

Time series cross-validation



- Forecast accuracy averaged over test sets.
- Also known as “evaluation on a rolling forecasting origin”

Creating the rolling training sets

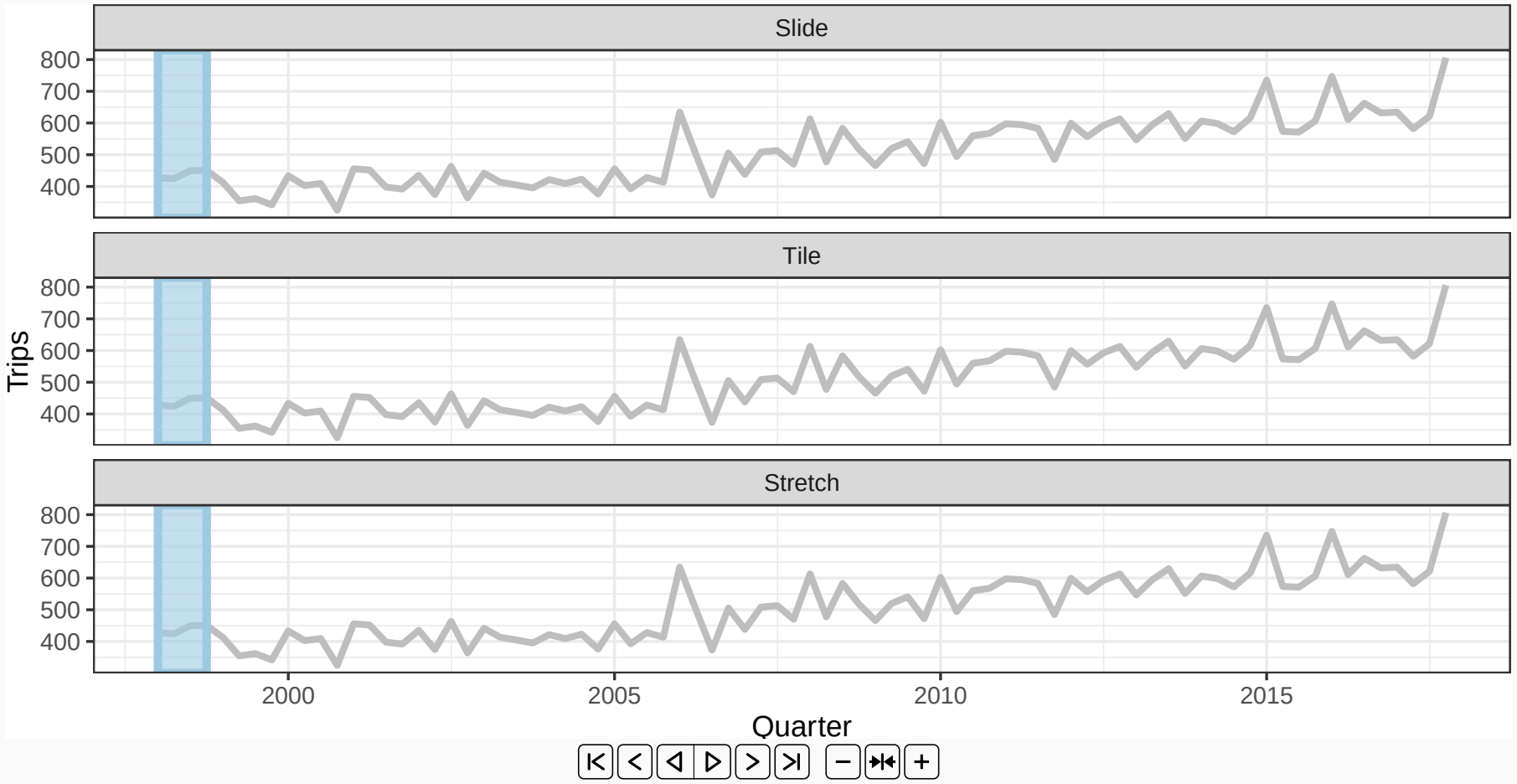
There are three main rolling types which can be used.

- Stretch: extends a growing length window with new data.
- Slide: shifts a fixed length window through the data.
- Tile: moves a fixed length window without overlap.

Three functions to roll a tsibble: `stretch_tsibble()`, `slide_tsibble()`, and `tile_tsibble()`.

For time series cross-validation, stretching windows are most commonly used.

Creating the rolling training sets



Time series cross-validation

Stretch with a minimum length of 3, growing by 1 each step.

```
forecast_horizon <- 8
google_2015_tr <- google_2015 %>%
  slice(1:(n()-forecast_horizon)) %>%
  stretch_tsibble(.init = 3, .step = 1)
```

```
## # A tsibble: 29,887 x 3 [1]
## # Key:       .id [242]
##      day Close   .id
##    <int> <dbl> <int>
## 1   253  522.      1
## 2   254  511.      1
## 3   255  499.      1
## 4   253  522.      2
## # ... with 2.988e+04 more rows
```

Time series cross-validation

Estimate RW w/ drift models for each window.

```
google_fit_tr <- google_2015_tr %>%
  model(drift=RW(Close ~ drift()))
```

```
## # A mable: 242 x 3
## # Key:     .id, Symbol [242]
##     .id Symbol drift
##   <int> <chr>  <model>
## 1     1 GOOG   <RW w/ drift>
## 2     2 GOOG   <RW w/ drift>
## 3     3 GOOG   <RW w/ drift>
## 4     4 GOOG   <RW w/ drift>
## # ... with 238 more rows
```

Time series cross-validation

Produce 8 step ahead forecasts from all models.

```
google_fc_tr <- google_fit_tr %>%
  forecast(h=8) %>%
  group_by(.id) %>%
  mutate(h=row_number()) %>%
  ungroup()
```

```
## # A fable: 1,936 x 7 [1]
## # Key:     .id, Symbol, .model [242]
##     .id Symbol .model   day Close .distribution
##   <int> <chr>  <chr>  <int> <dbl> <dist>
## 1     1 GOOG   drift    256  488. N(488, 0.47)
## 2     1 GOOG   drift    257  476. N(476, 1.40)
## 3     1 GOOG   drift    258  465. N(465, 2.79)
## 4     1 GOOG   drift    259  454. N(454, 4.65)
```

Time series cross-validation

```
# Cross-validated
google_fc_tr %>% accuracy(google_2015)
# Training set
google_2015 %>% model(NAIVE(Close)) %>% accuracy()
```

	RMSE	MAE
Cross-validation	25.08698	16.178139
Training	11.18958	7.127985

Outline

Winkler score

Winkler proposed a scoring method to enable comparisons between prediction intervals:

- it takes account of both coverage and width of the intervals.

Winkler score

$$W(l_t, u_t, y_t) = \begin{cases} u_t - l_t & \text{if } l_t < y_t < u_t \\ (u_t - l_t) + \frac{2}{\alpha}(l_t - y_t) & \text{if } y_t < l_t \\ (u_t - l_t) + \frac{2}{\alpha}(y_t - u_t) & \text{if } y_t > u_t \end{cases}$$

Prediction interval accuracy

```
 # Compute interval accuracy
google_fc_tr %>%
 accuracy(google_2015,
  measures = interval_accuracy_measures) %>%
  mutate(Method = paste(.model, "method")) %>%
 select(Method, winkler) %>%
 gt::gt("Method") %>%
 gt::as_latex()
```

	winkler
drift method	167.8425

Outline

Lab session 6

Compute seasonal naïve forecasts for daily A&E n_attendance:

1. use `slice()` function to subset data into train and test
   - keep the last 42 days for test set
2. Speicify model and train data on train set
3. visualise forecasts
4. Test if the residuals are white noise.
   - use `gg_tsdisplay` function and `Lj` test
   - What do you conclude?

Lab session 6

5. Create folds/windows for time series cross validation
   - **Hint:** use `stretch_tsibble(.init = 4*365, .step = 1)`
6. Train model on each fold/window
7. Forecast for 42 days
8. Compute RMSE and MAE to evaluate point forecast
9. Evaluate the prediction intervals using Winkler score.

Recap

1. First, import your data and prepare them using `tsibble` function.
2. Visualise and see wether your series contains key features
3. Determine how much of your data you want to allocate to training, and how much to testing; the sets should not overlap.
4. Subset the data to create a training set, which you will use as an argument in your forecasting function(s). Optionally, you can also create a test set to use later.
5. Compute forecasts of the training set using whichever forecasting function(s) you choose, and set h equal to the number of values you want to forecast.

Recap

6. Use residual diagnostic based on residuals in the training set to see wether all informations is captured by models.
7. Create different windows to evaluate forecast accuracy using time series cross validation
8. Train model to each window
9. To view the results of accuracy, use the accuracy() function with the `fable` as the first argument and original data as the second.
10. Pick a measure in the output to evaluate the forecast(s); a smaller error indicates higher accuracy.
11. Forecast using all data for test set and visualise forecasts against actual values
12. Finally, produce forecast using the selected approach for future.