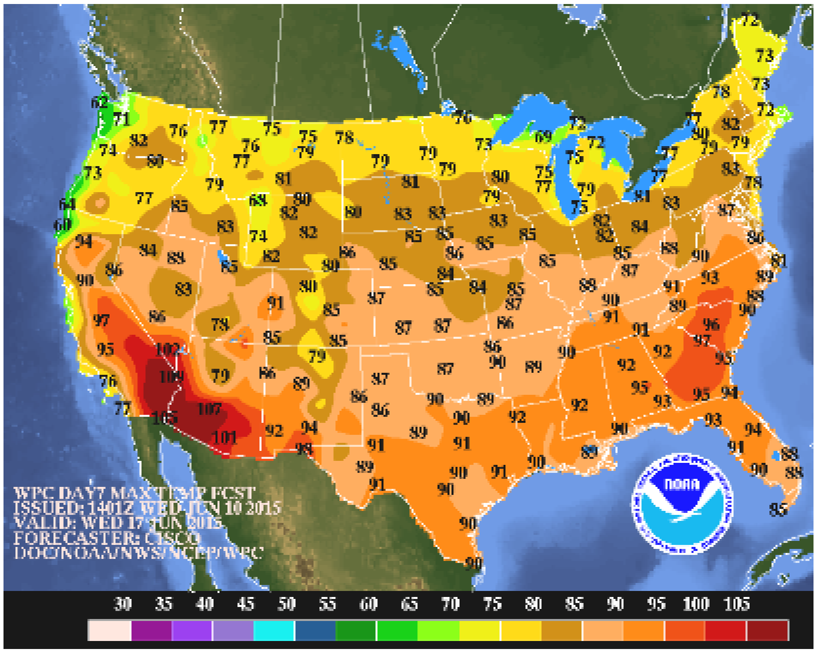
Errors in Weather Forecasting

Hypothesis: Errors in weather forecasting are decreasing and there is very little systematic error.

Data: Historical Weather forecasts for various locations in the continental United States. Data will be predicted daily high and low temperatures and predicted probability of precipitation. Data will vary by how many days ahead the forecast is for. Data will also include the actual weather data for each forecasted date and location.

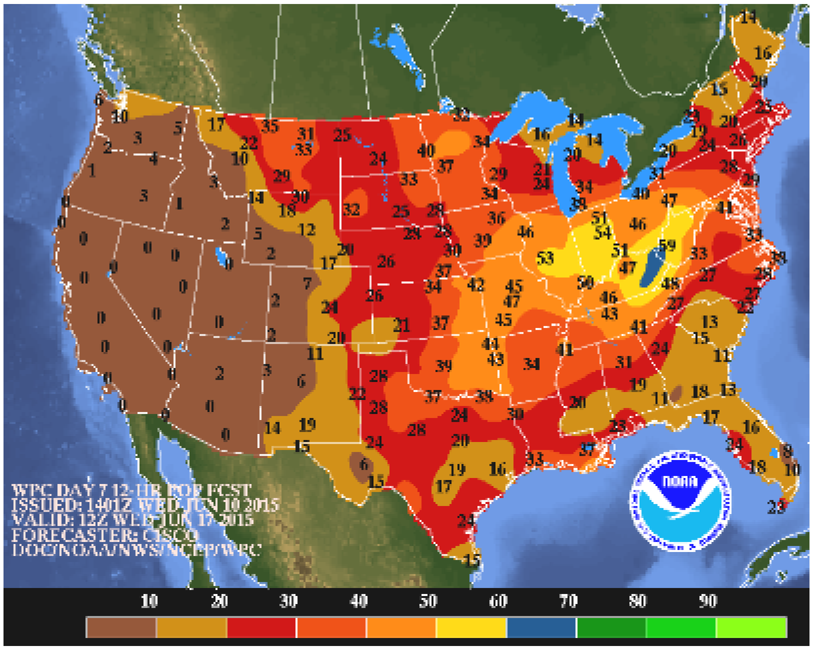
Obtaining Historical Weather forecasts: The archived weather forecasts were available but only in graphic form (.GIF image files) within a map of the United States. The numbers were embedded in the graphic of medium range forecasts (3 – 7 days ahead).

Maximum temperature for 6/17/15 predicted 7 days before



DAY7_MAX_2015061012_filled.gif

Probability of Precipitation for 6/17/15 predicted 7 days before



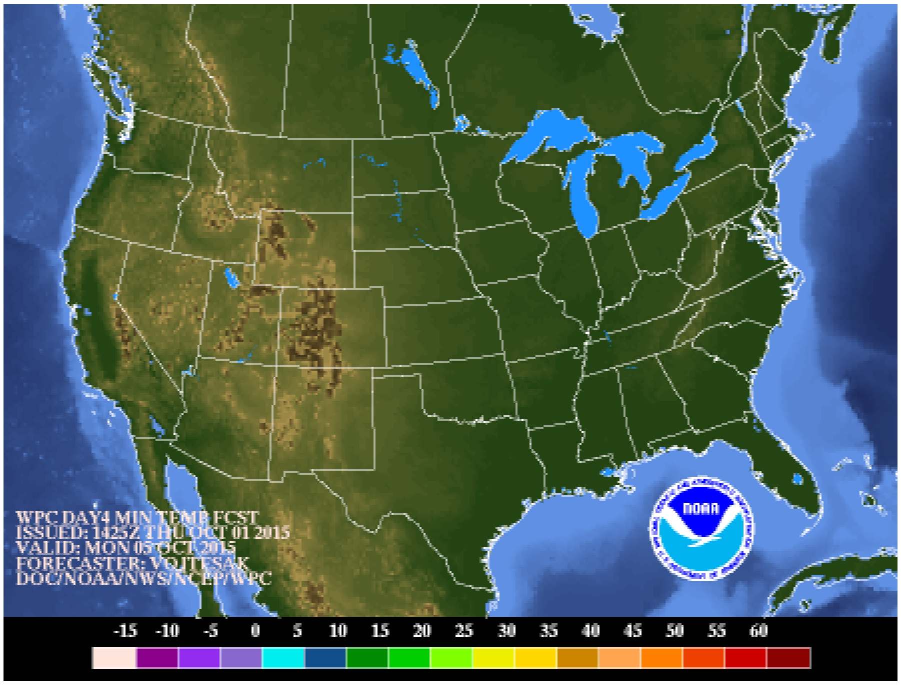
DAY7_POP1_2015061012_filled.gif

The numbers contained up to 3 characters from a set of 11: 0, 1, 2,…,9, -. The character font and color (black) was always the same. There was no anti-aliasing, and the characters overwrote all other colors. The locations of the numbers, for a given number of characters comprising it, was always the same. The strategy was to search for the character pixel patterns making up the number at each location.

```
((1,1,1,1,1,1,1,1,0), # 2 upside down
 (1,1,1,1,1,1,1,1,0),
 (0,1,1,0,0,0,1,1,0),
 (0,0,1,1,0,0,0,1,0),
 (0,0,0,1,1,0,0,0,0),
 (0,0,0,0,1,1,0,0,0),
 (0,0,0,0,0,1,1,0,0),
 (0,0,0,0,0,1,1,1,0),
 (0,0,0,0,0,1,1,1,0),
 (1,0,0,0,0,1,1,1,0),
 (1,1,0,0,1,1,1,1,0),
 (0,1,1,1,1,1,1,0,0),
 (0,0,1,1,1,1,0,0,0)),
```

The .GIF image files were systematically named and publicly available. They were downloaded in a Python script (download_images.py) using the requests module via http and then saved as a binary file.

Missing Data: Some of the graphics were missing numbers:



2015100112_DAY4_MIN_filled.gif

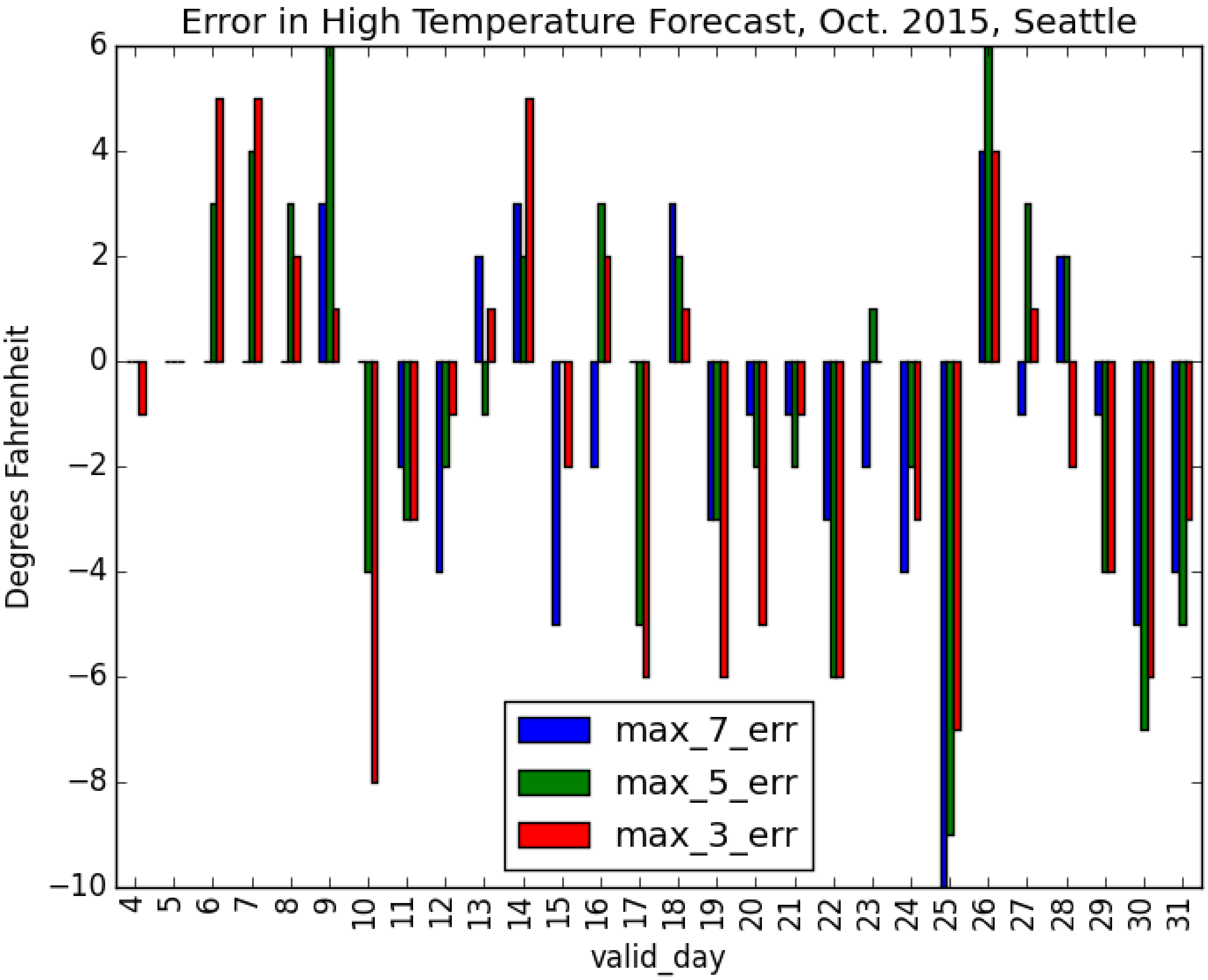
Converting .GIF images to .BMP images: The .GIF are a compressed format which makes it harder to work with the pixel data. So the files were converted to the uncompressed .BMP format with a python script (convert_gif_to_bmp.py) using the PIL module via the Image.open and Image.save methods. Fortuitously, it was quite straightforward.

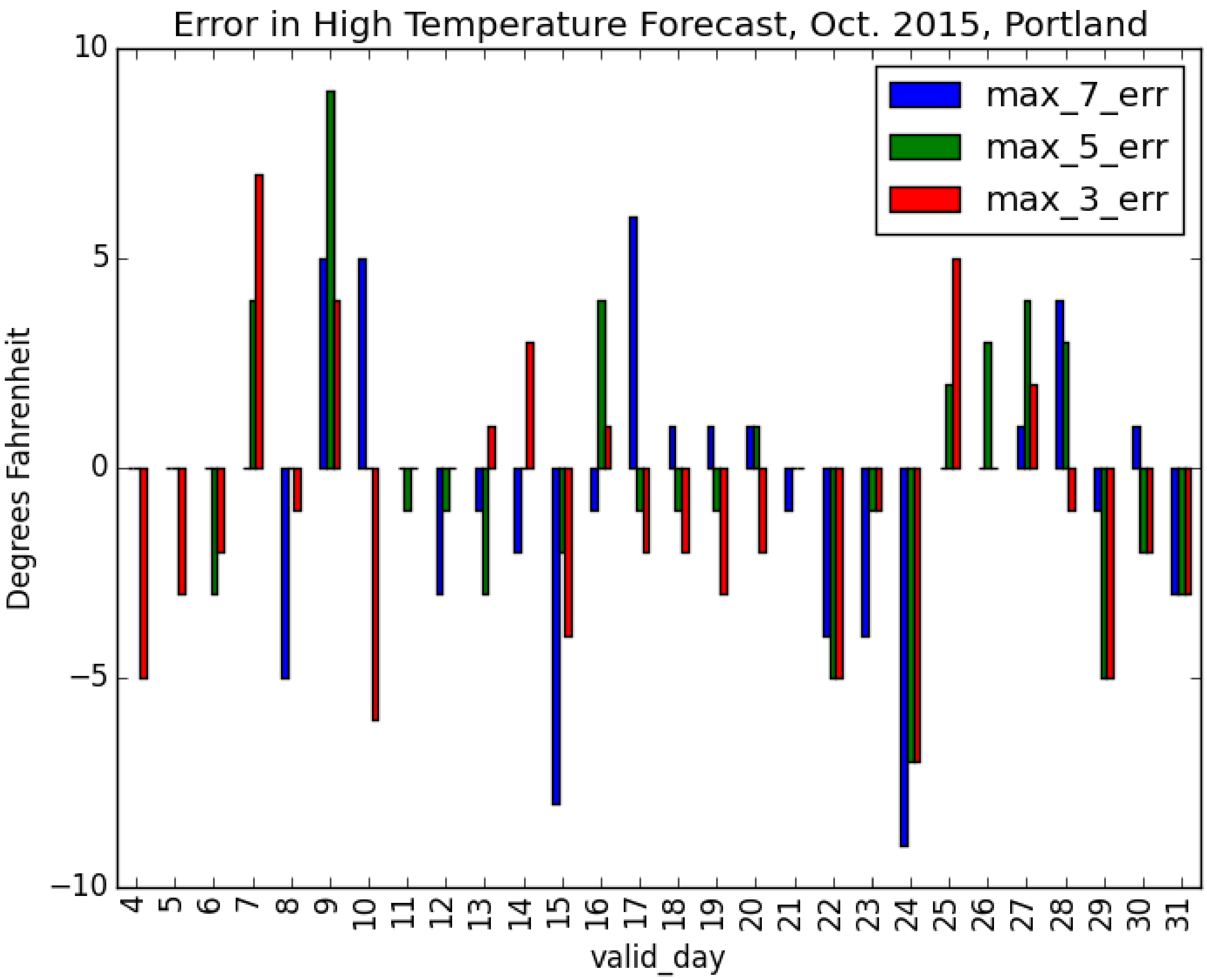
Extracting the forecasted numbers: The embedded numbers were extracted with a Python script (extract_numbers.py) by interrogating the header of the image file and locating the memory addresses for the numbers corresponding to each location in the map. The issue date of the forecast, its type (e.g. max temperature) and forecast period (days) was contained in the file name. The data was written to a text (med_range_forecast.csv) file. One simple calculation was made and added to the data file (date forecast is valid for).

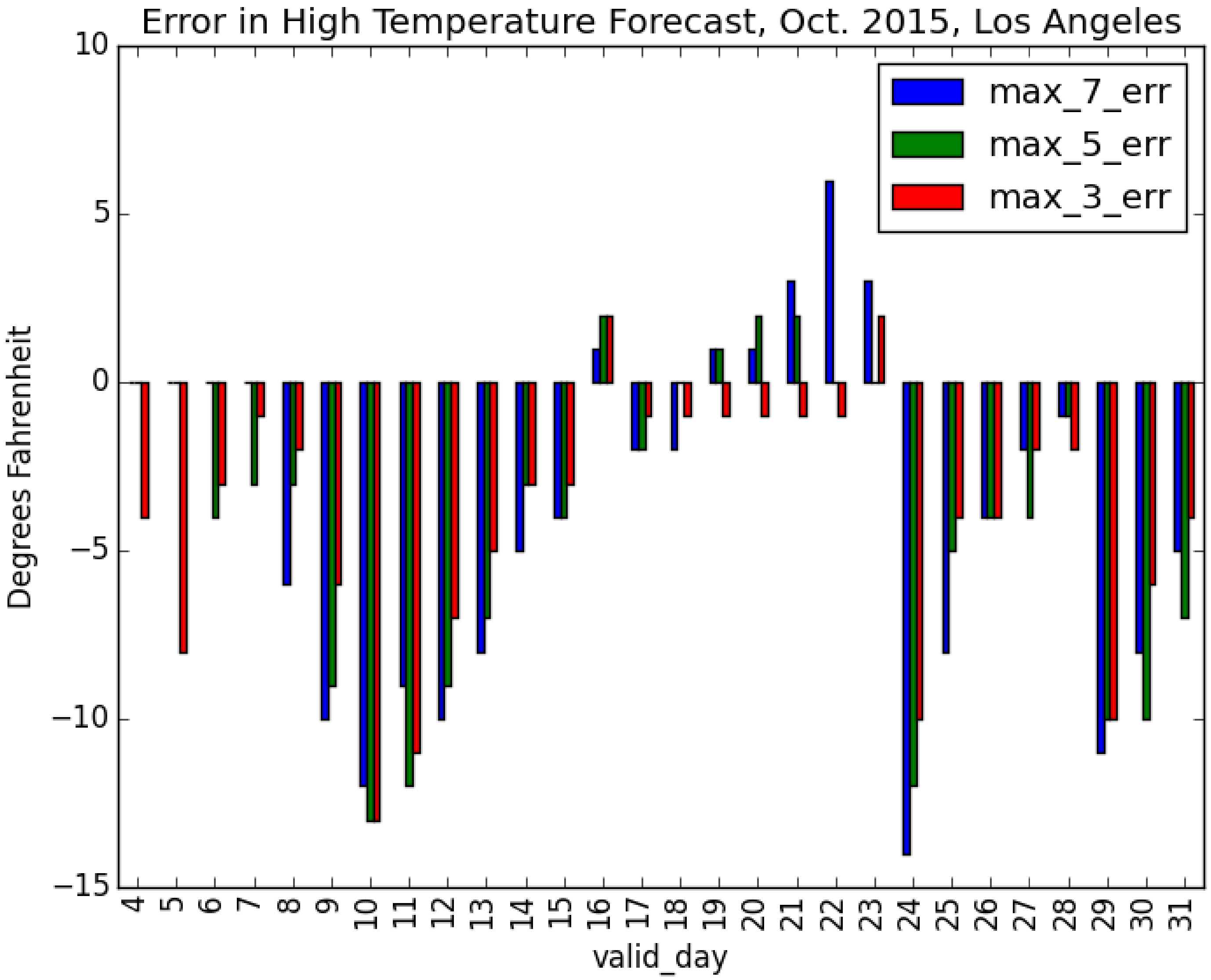
Obtaining the observed values: The observed weather data (actual values on the day) were obtained from archived web pages and stored in a text file (past_observation.csv).

Merging Forecast Data with Observed Data: The forecasted data had many lines for each line in the observed weather data (a location, a date, and observer min/max temperatures, and precipitation). So the forecasted data was pivoted in a Python script (pivot_and_join.py) to a wide format with one line of data for a given location and date. The data was previously switched from the issue date to the valid date (the date for which the forecast is valid for). All this data was stored in a text file (med_range_forecast_obs.csv).

Analyzing the errors in the forecasts: A simple subtraction for the temperature data yields the errors in the forecasts. The values were plotted:




Error in High Temperature Forecast, Oct. 2015, Portland
Degrees Fahrenheit
10
5
0
−5
−10
valid_day
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
max_7_err
max_5_err
max_3_err



The average (mean) of the errors was calculated:

```
Calculate mean (signed) error for Seattle...
max_7_err   -1.291667
max_6_err   -0.880000
max_5_err   -0.769231
max_4_err   -1.074074
max_3_err   -1.321429

Calculate mean (signed) error for Portland...
max_7_err   -0.708333
max_6_err   -0.120000
max_5_err   -0.230769
max_4_err   -0.555556
```

```
max_3_err   -1.107143

Calculate mean (signed) error for Los Angeles...
max_7_err   -4.416667
max_6_err   -4.280000
max_5_err   -4.423077
max_4_err   -4.444444
max_3_err   -3.928571
```

For Seattle and especially Los Angeles the forecasts seem to be quite low for this time period.