

## Background and motivation

While global warming is known to have already increased temperatures around the globe, with that trend expected to continue and even amplify in the future, less is known about changes in temperature variability. Variability is important for many organisms, where large fluctuations in temperature can be damaging. Here in the Bay we are well-versed in the requirements of large temperature variability not only week to week, but even over the course of a single day (wear shorts but still pack a sweater anyone?). If global warming affects temperature evenly, whether it's average or the extremes, the temperature variability will remain unchanged. However, if it heats lower temperature extremes more than it heats high temperatures, you would expect the distribution of temperature to narrow, and conversely if it affects extreme highs more you would expect increased variability, along with a shifting mean.

Here I attempt to understand what factors affect temperature variability around the globe using monthly temperature measurements compiled by NOAA. There are many measures of variability, perhaps the most obvious being standard deviation, but I use the simple metric of the difference between observed monthly maximum and minimum temperatures. This spread provides a proxy for temperature variability, with larger spreads suggesting higher variability.

I consider various features that may be related to temperature variability, such as latitude, elevation, average temperature, and season. I also look at potential temporal trends.

## Setting up BigQuery and dependencies

```
# Run this cell to authenticate yourself to BigQuery
from google.colab import auth
auth.authenticate_user()
project_id = # INSERT YOUR PROJECT ID HERE

# Initialize BigQuery client
from google.cloud import bigquery
client = bigquery.Client(project=project_id) # pass in your projectid

import altair as alt
import matplotlib as mpl

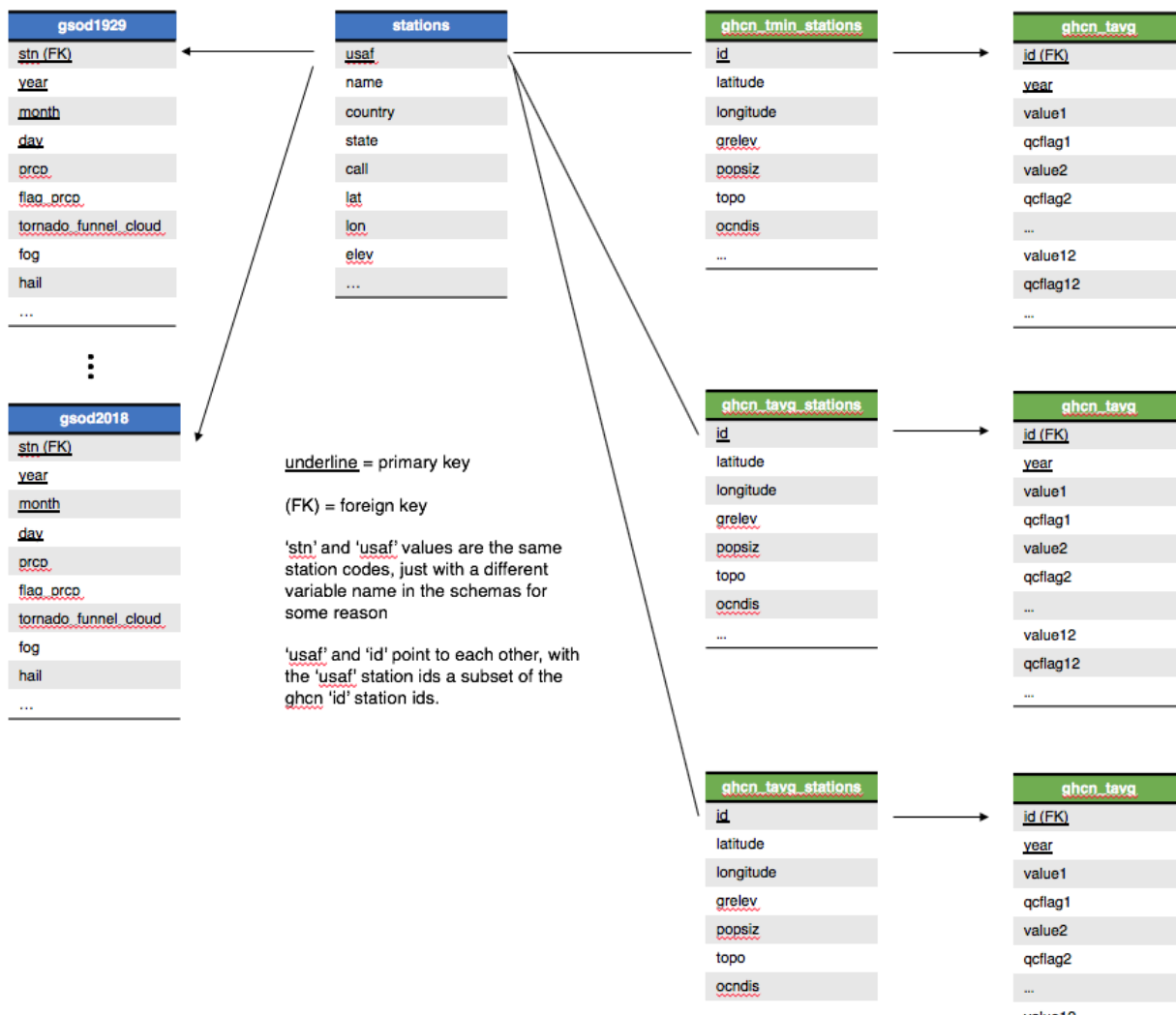
# Run this cell to create a dataset to store your model
model_dataset_name = 'project3'
dataset = bigquery.Dataset(client.dataset(model_dataset_name))
dataset.location = 'US'
client.create_dataset(dataset)
```

I use two datasets from BigQuery, one for monthly data and one with daily data and some weather station attributes. The daily data covers only weather stations within the World Meteorological Organization set of stations, while the monthly data includes those as well as a few thousand other stations. Both are set up as relational databases.

The two datasets (NOAA monthly or NOAA daily) are related to each other via the weather station id key in the '\_stations' databases, called 'id' in monthly data and having '000' appended to the end and 3 digits appended to the front of the 5-digit name, and the 5-digit 'usaf' id for the daily data. The stations tables include characteristics about each station, some of which is repeated in each database. There were a few station attributes of potential interest to me in my model, such as land cover (desert, forest, urban, etc.) but for the majority of stations these were NULL, so I did not include it as a candidate covariate.

Within each dataset are separate databases containing the weather data. For the daily data, each year has its own database with {station id, year, month, day} as the key, pointing to the stations table where for some reason 'str' and 'usaf' have different names but both refer to the 5-digit unique station identifier. For the monthly data, instead each variable (tmax, tmin, tavg) is separated into its own database with station {id, year} as its key. The monthly values are attributes and include separate attributes for each month indicating whether there are e.g. any data flags. These are linked to the 'stations' table via the 'id' (See diagram). Note I removed any data with a data flag. In both datasets, there are large amounts of missing values (NULL) since there is an entry as long as there is a value for at least one of the attributes. For example, there may be a row for 1980-01-01 with a value for precipitation, but NULL for all other attributes.

---



## ▼ Clean up monthly temperature data and calculate tmax-tmin spread

Each row in the original table (separate tables for tmax, tmin, tavg) has a station-year key with columns corresponding to the values in each month and some quality control information for each monthly measurement in other columns. Here I wrangle the data into a new table with a station-year-month key, with separate columns for tmax, tmin, their difference, and tavg, making sure to filter out any values that have a quality control flag or that are missing.

Because this code is lengthy, I save this as a table 'project3.results\_20181130' and read it in later as needed 'project3.results\_20181130'.

```
%%bigquery --project $project_id
## There may be a more efficient way to do this, but it works and is fast...
## Temperature values are divided by 100 to convert to deg C

## January
SELECT a.id, a.year, a.value1/100 AS tmax, b.value1/100 AS tmin, ROUND(a.value1/100-b.value1/100, 2) AS spread, '1' AS month, c.value1/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value1 != (-9999) AND b.value1 != (-9999) AND c.value1 != (-9999) ## Filter out missing values
AND a.qcflag1='' AND b.qcflag1='' AND c.qcflag1='' ## Empty QC flags is a good thing
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## February
SELECT a.id, a.year, a.value2/100 AS tmax, b.value2/100 AS tmin, ROUND(a.value2/100-b.value2/100, 2) AS spread, '2' AS month, c.value2/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value2 != (-9999) AND b.value2 != (-9999) AND c.value2 != (-9999)
AND a.qcflag2='' AND b.qcflag2='' AND c.qcflag2=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## March
SELECT a.id, a.year, a.value3/100 AS tmax, b.value3/100 AS tmin, ROUND(a.value3/100-b.value3/100, 2) AS spread, '3' AS month, c.value3/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value3 != (-9999) AND b.value3 != (-9999) AND c.value3 != (-9999)
AND a.qcflag3='' AND b.qcflag3='' AND c.qcflag3=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980
```

```

UNION ALL

## April
SELECT a.id, a.year, a.value4/100 AS tmax, b.value4/100 AS tmin, ROUND(a.value4/100-b.value4/100, 2) AS spread, '4' AS month, c.value4/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value4 != (-9999) AND b.value4 != (-9999) AND c.value4 != (-9999)
AND a.qcflag4='' AND b.qcflag4='' AND c.qcflag4=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## May
SELECT a.id, a.year, a.value5/100 AS tmax, b.value5/100 AS tmin, ROUND(a.value5/100-b.value5/100, 2) AS spread, '5' AS month, c.value5/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value5 != (-9999) AND b.value5 != (-9999) AND c.value5 != (-9999)
AND a.qcflag5='' AND b.qcflag5='' AND c.qcflag5=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## June
SELECT a.id, a.year, a.value6/100 AS tmax, b.value6/100 AS tmin, ROUND(a.value6/100-b.value6/100, 2) AS spread, '6' AS month, c.value6/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value6 != (-9999) AND b.value6 != (-9999) AND c.value6 != (-9999)
AND a.qcflag6='' AND b.qcflag6='' AND c.qcflag6=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## July
SELECT a.id, a.year, a.value7/100 AS tmax, b.value7/100 AS tmin, ROUND(a.value7/100-b.value7/100, 2) AS spread, '7' AS month, c.value7/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value7 != (-9999) AND b.value7 != (-9999) AND c.value7 != (-9999)
AND a.qcflag7='' AND b.qcflag7='' AND c.qcflag7=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## August
SELECT a.id, a.year, a.value8/100 AS tmax, b.value8/100 AS tmin, ROUND(a.value8/100-b.value8/100, 2) AS spread, '8' AS month, c.value8/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value8 != (-9999) AND b.value8 != (-9999) AND c.value8 != (-9999)
AND a.qcflag8='' AND b.qcflag8='' AND c.qcflag8=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## September
SELECT a.id, a.year, a.value9/100 AS tmax, b.value9/100 AS tmin, ROUND(a.value9/100-b.value9/100, 2) AS spread, '9' AS month, c.value9/100
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value9 != (-9999) AND b.value9 != (-9999) AND c.value9 != (-9999)
AND a.qcflag9='' AND b.qcflag9='' AND c.qcflag9=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## October
SELECT a.id, a.year, a.value10/100 AS tmax, b.value10/100 AS tmin, ROUND(a.value10/100-b.value10/100, 2) AS spread, '10' AS month, c.value
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value10 != (-9999) AND b.value10 != (-9999) AND c.value10 != (-9999)
AND a.qcflag10='' AND b.qcflag10='' AND c.qcflag10=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## November
SELECT a.id, a.year, a.value11/100 AS tmax, b.value11/100 AS tmin, ROUND(a.value11/100-b.value11/100, 2) AS spread, '11' AS month, c.value
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value11 != (-9999) AND b.value11 != (-9999) AND c.value11 != (-9999)
AND a.qcflag11='' AND b.qcflag11='' AND c.qcflag11=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

UNION ALL

## December
SELECT a.id, a.year, a.value12/100 AS tmax, b.value12/100 AS tmin, ROUND(a.value12/100-b.value12/100, 2) AS spread, '12' AS month, c.value
FROM `bigquery-public-data.ghcn_m.ghcnm_tmax` a,
`bigquery-public-data.ghcn_m.ghcnm_tmin` b,
`bigquery-public-data.ghcn_m.ghcnm_tavg` c
WHERE a.value12 != (-9999) AND b.value12 != (-9999) AND c.value12 != (-9999)
AND a.qcflag12='' AND b.qcflag12='' AND c.qcflag12=''
AND a.id=b.id AND a.year=b.year AND a.id=c.id AND a.year=c.year AND a.year > 1980

```


24	42500188000	1981	3.25	-5.45	8.70	1	-1.32
25	42500367931	1981	-1.10	-11.37	10.27	1	-6.20
26	42500489770	1981	7.28	-7.59	14.87	1	0.00
27	42500470991	1981	-4.51	-17.94	13.43	1	-11.12
28	42500457267	1981	6.45	0.12	6.33	1	3.25
29	42500358997	1981	3.08	-3.56	6.64	1	-0.11
...	...	...	...	...	...	...	...
1016140	42500102845	2017	2.32	-3.38	5.70	12	-0.52
1016141	42500475255	2017	-6.05	-14.21	8.16	12	-10.12
1016142	42500046826	2017	15.69	0.39	15.30	12	8.04
1016143	42500313969	2017	10.83	-1.93	12.76	12	4.45
1016144	42500051528	2017	10.67	-6.86	17.53	12	1.91
1016145	42500124181	2017	1.89	-6.76	8.65	12	-2.43
1016146	42500213303	2017	-8.23	-16.80	8.57	12	-12.51
1016147	42500404561	2017	10.46	-1.11	11.57	12	4.68
1016148	42500355362	2017	5.02	-1.62	6.64	12	1.70
1016149	42500147093	2017	7.90	-8.80	16.70	12	-0.44
1016150	42500034756	2017	11.64	0.97	10.67	12	6.31
1016151	42500427559	2017	5.32	-6.42	11.74	12	-0.54
1016152	42500380165	2017	13.02	3.06	9.96	12	8.04
1016153	42500427714	2017	7.02	-11.12	18.14	12	-2.04
1016154	42500243751	2017	-1.38	-10.60	9.22	12	-5.98
1016155	42500476827	2017	0.16	-8.85	9.01	12	-4.34
1016156	42500295960	2017	10.13	-1.92	12.05	12	4.11
1016157	42500292848	2017	15.92	-0.63	16.55	12	7.65
1016158	42500425733	2017	7.41	-7.27	14.68	12	0.07
1016159	42500242409	2017	-0.82	-12.77	11.95	12	-6.79
1016160	42500402108	2017	9.69	-0.86	10.55	12	4.42
1016161	42500381770	2017	13.13	3.01	10.12	12	8.07

## ▼ Exploratory data analysis

I begin by cleaning and organizing the data. I then present plots for each candidate covariate's relation to temperature spread and in some cases some potential transformations of that covariate as well.

## ▼ Response variable 'spread' and available stations

```
%%bigquery --project $project_id
SELECT COUNT(DISTINCT id) n_stations
FROM `project3.results_20181130`
```



n_stations
0
4870

```
%%bigquery --project $project_id
SELECT COUNT(DISTINCT id) n_stations
## Don't worry about the covariates below for now, just getting # of stations
FROM
`project3.results_20181130` a,
(SELECT DISTINCT(stn)
FROM `bigquery-public-data.noaa_gsod.gsod20*`
WHERE prcp != 99.99 #Note those with precip also have the tornado data, so I don't include code for that here
) c,
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200*`
GROUP BY stn
) d
WHERE
# Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the WMO dataset
AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
AND c.stn = d.stn
```



n_stations
0
2413

There are 4,870 stations with at least 1 measurement in the dataset. However, we will be examining a restricted set of 2,413 sites, those overseen by the World Meteorological Organization, in order to incorporate some additional covariates (more on that later).

## ▼ Compile cleaned data with response variable candidate covariates

I put the final cleaned dataset I came to below. This includes information from the monthly GHCN data tables as well as the daily NOAA GSOD data tables on BigQuery.

```
%%bigquery --project $project_id
## Takes ~1min to run
## 'id' is a station identifier
SELECT
  a.id,
  spread,
  a.year,
  a.month,
  CASE WHEN (a.month IN ('12','1','2') AND latitude>=0)
    OR (a.month IN ('6','7','8') AND latitude<0) then 'winter'
  WHEN (a.month IN ('3','4','5') AND latitude>=0)
    OR (a.month IN ('9','10','11') AND latitude<0) then 'spring'
  WHEN (a.month IN ('6','7','8') AND latitude>=0)
    OR (a.month IN ('12','1','2') AND latitude<0) then 'summer'
  WHEN (a.month IN ('9','10','11') AND latitude>=0)
    OR (a.month IN ('3','4','5') AND latitude<0) then 'fall'
  END AS season,
  longitude,
  latitude,
  CASE WHEN ABS(latitude) BETWEEN 0 AND 30 then 'tropics'
  WHEN ABS(latitude) > 30 AND latitude <=60 then 'midlatitudes'
  WHEN ABS(latitude) > 60 AND latitude <=90 then 'polar'
  END AS lat_bin,
  tavg,
  CASE WHEN grelev < 5 then '0-5m'
  WHEN grelev >=5 AND grelev < 25 then '5-25m'
  WHEN grelev >=25 AND grelev < 100 then '25-100m'
  WHEN grelev >=100 AND grelev < 500 then '100-500m'
  WHEN grelev >=500 AND grelev < 1000 then '500-1000m'
  WHEN grelev >=100 AND grelev < 2000 then '1000-2000m'
  WHEN grelev >=2000 then '>=2000m'
  END AS elevation_bin,
  LOG(CASE WHEN grelev !=0 THEN grelev ELSE 1 END) AS log_elevation, # Tweak to adjust for log(0)
  prcp_mean_annual,
  LOG(CASE WHEN prcp_mean_annual !=0 THEN prcp_mean_annual ELSE 1 END) AS log_prcp_mean_annual, # Tweak to adjust for log(0)
  tornadoes

#-----
FROM
  ## Saved table (response variable, year, month, tavg)
  `project3.results_20181130` a,
  ## Attributes of the station (e.g. elevation)
  `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
  ## Average 2000-2018 annual precipitation
  (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
  FROM (
    SELECT stn, SUM(prcp) AS prcp_annual, year
    FROM `bigquery-public-data.noaa_gsod.gsod20*`
    WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
    GROUP BY year,stn
  )
  GROUP BY stn
) c,
  ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
  (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
  FROM `bigquery-public-data.noaa_gsod.gsod200*`
  GROUP BY stn
) d

#-----
WHERE
  a.id=b.id AND grelev IS NOT NULL
  # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
  AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
  AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
  AND c.stn = d.stn
```



	id	spread	year	month	season	latitude	lat_bin	tavg	elevation_bin	log_elevation	prcp_mean_annual	log_prcp
0	50194304000	7.50	1981	1	summer	-20.82	tropics	30.10	5-25m	2.639057		0.0
1	50194304000	9.50	1982	1	summer	-20.82	tropics	29.80	5-25m	2.639057		0.0
2	50194304000	9.40	1983	1	summer	-20.82	tropics	30.40	5-25m	2.639057		0.0
3	50194304000	9.40	1984	1	summer	-20.82	tropics	30.60	5-25m	2.639057		0.0
4	50194304000	9.90	1985	1	summer	-20.82	tropics	29.50	5-25m	2.639057		0.0
5	50194304000	8.50	1986	1	summer	-20.82	tropics	29.50	5-25m	2.639057		0.0
6	50194304000	8.20	1987	1	summer	-20.82	tropics	28.60	5-25m	2.639057		0.0
7	50194304000	10.00	1988	1	summer	-20.82	tropics	30.20	5-25m	2.639057		0.0
8	50194304000	8.10	1989	1	summer	-20.82	tropics	30.00	5-25m	2.639057		0.0
9	50194304000	6.10	1981	2	summer	-20.82	tropics	28.60	5-25m	2.639057		0.0
10	50194304000	8.20	1982	2	summer	-20.82	tropics	29.00	5-25m	2.639057		0.0
11	50194304000	8.70	1983	2	summer	-20.82	tropics	30.80	5-25m	2.639057		0.0
12	50194304000	8.30	1984	2	summer	-20.82	tropics	30.60	5-25m	2.639057		0.0
13	50194304000	8.80	1985	2	summer	-20.82	tropics	30.40	5-25m	2.639057		0.0
14	50194304000	9.20	1986	2	summer	-20.82	tropics	29.80	5-25m	2.639057		0.0
15	50194304000	8.10	1987	2	summer	-20.82	tropics	29.90	5-25m	2.639057		0.0
16	50194304000	7.50	1989	2	summer	-20.82	tropics	29.90	5-25m	2.639057		0.0
17	50194304000	8.50	1981	3	fall	-20.82	tropics	29.50	5-25m	2.639057		0.0
18	50194304000	9.00	1982	3	fall	-20.82	tropics	29.00	5-25m	2.639057		0.0
19	50194304000	8.30	1983	3	fall	-20.82	tropics	29.10	5-25m	2.639057		0.0
20	50194304000	8.10	1984	3	fall	-20.82	tropics	29.60	5-25m	2.639057		0.0
21	50194304000	9.30	1985	3	fall	-20.82	tropics	30.60	5-25m	2.639057		0.0

The table above contains individual values at each location for each year-month combination with data. Below I aggregate some of the attributes over time so that each row corresponds to a different station with various attributes of it in the columns. This is used for the exploratory data analysis that follows.

```

%%bigquery --project $project_id station_attributes

SELECT id, AVG(spread) avg_spread, AVG(tavg) tmean, AVG(longitude) long, AVG(latitude) lat, lat_bin, AVG(elevation) AS elev, AVG(log_elevation) AS log_elev, AVG(prcp_mean_annual) AS prcp_mean_annual, AVG(log_prcp_mean_annual) AS log_prcp_mean_annual, AVG(tornadoes) AS tornadoes
FROM (
  SELECT
    a.id,
    spread,
    a.year,
    CASE WHEN a.year < 1995 then 'pre-1995'
          WHEN a.year >=1995 then '1995-2018'
    END AS year_bin,
    a.month,
    CASE WHEN (a.month IN ('12','1','2') AND latitude>=0)
           OR (a.month IN ('6','7','8') AND latitude<0) then 'winter'
           WHEN (a.month IN ('3','4','5') AND latitude>=0)
           OR (a.month IN ('9','10','11') AND latitude<0) then 'spring'
           WHEN (a.month IN ('6','7','8') AND latitude>=0)
           OR (a.month IN ('12','1','2') AND latitude<0) then 'summer'
           WHEN (a.month IN ('9','10','11') AND latitude>=0)
           OR (a.month IN ('3','4','5') AND latitude<0) then 'fall'
    END AS season,
    longitude,
    latitude,
    CASE WHEN ABS(latitude) BETWEEN 0 AND 10 then 'tropics'
           WHEN ABS(latitude) > 10 AND latitude <=40 then 'extratropics'
           WHEN ABS(latitude) > 40 AND latitude <=60 then 'midlatitudes'
           WHEN ABS(latitude) > 60 AND latitude <=90 then 'polar'
    END AS lat_bin,
    tavg,
    CASE WHEN grelev < 5 then '0-5m'
           WHEN grelev >=5 AND grelev < 25 then '5-25m'
           WHEN grelev >=25 AND grelev < 100 then '25-100m'
           WHEN grelev >=100 AND grelev < 500 then '100-500m'
           WHEN grelev >=500 AND grelev < 1000 then '500-1000m'
           WHEN grelev >=1000 AND grelev < 2000 then '1000-2000m'
           WHEN grelev >=2000 then '>=2000m'
    END AS elevation_bin,
    LOG(CASE WHEN grelev !=0 THEN grelev ELSE 1 END) AS log_elevation, # Tweak to adjust for log(0),
    grelev AS elevation,
    prcp_mean_annual,
    LOG(CASE WHEN prcp_mean_annual !=0 THEN prcp_mean_annual ELSE 1 END) AS log_prcp_mean_annual, # Tweak to adjust for log(0)
    tornadoes
  )
#-----
FROM
  ## Saved table (response variable, year, month, tavg)
  `project3.results_20181130` a,
  ## Attributes of the station (e.g. elevation)
  `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
  ## Average 2000-2018 annual precipitation
  `project3.results_20181130` c

```

```

(SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
FROM (
  SELECT stn, SUM(prcp) AS prcp_annual, year
  FROM `bigquery-public-data.noaa_gsod.gsod20*`
  WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
  GROUP BY year,stn
)
GROUP BY stn
) c,
## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200*`
GROUP BY stn
) d
#-----
WHERE
  a.id=b.id AND grelev IS NOT NULL
  # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
  AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
  AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
  AND c.stn = d.stn
)
GROUP BY id, elevation_bin, tornadoes, lat_bin

```

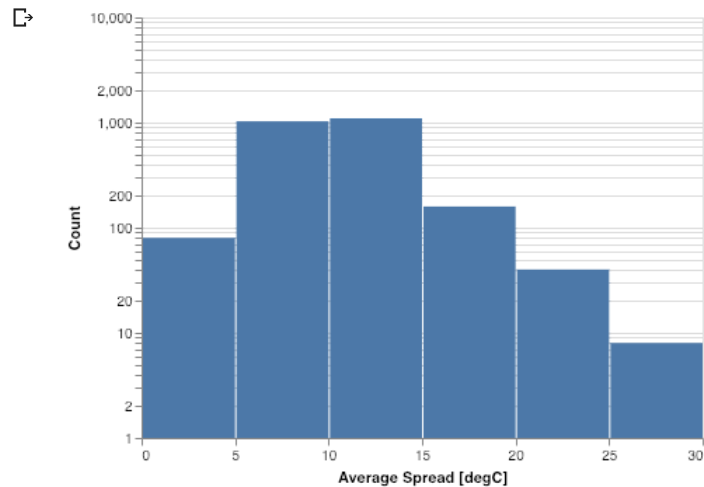
```
station_attributes.head(5)
```

	id	avg_spread	tmean	long	lat	lat_bin	elev	log_elev	elevation_bin	prcp_mean	log_precip	tornadoes
0	22220292000	4.527768	-12.667857	104.30	77.72	polar	0.0	0.0	0-5m	7.7	2.041220	no_tornadoes
1	42570219000	7.723744	0.082215	-161.80	60.78	polar	0.0	0.0	0-5m	20.6	3.025291	no_tornadoes
2	63401001000	3.560563	0.676995	-8.67	70.93	polar	0.0	0.0	0-5m	22.4	3.109061	no_tornadoes
3	22220674000	5.041921	-9.319272	80.40	73.50	polar	0.0	0.0	0-5m	14.6	2.681022	no_tornadoes
4	40371081000	7.038111	-11.319870	-81.25	68.78	polar	0.0	0.0	0-5m	4.5	1.504077	no_tornadoes

```

alt.Chart(station_attributes).mark_bar().encode(
  x=alt.X("avg_spread", bin=True, axis=alt.Axis(title="Average Spread [degC]")),
  y=alt.Y('count()', axis=alt.Axis(title="Count"), scale=alt.Scale(type='log'))
)

```



[Export as SVG](#)
[Export as PNG](#)
[View Source](#)
[Open in Vega Editor](#)

The histogram above shows average temperature spreads for each station. The majority of stations see an average spread of 5-15 degC, although there are a few outliers with very high swings.

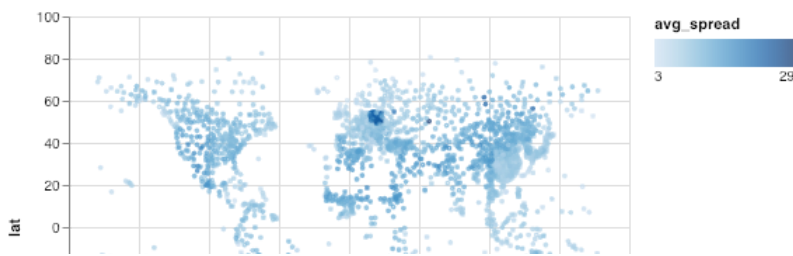
```

## Map of the average spread doesn't give too much insight
alt.Chart(station_attributes).mark_point(size=3).encode(
  x='long',
  y='lat',
  color='avg_spread'
)

```

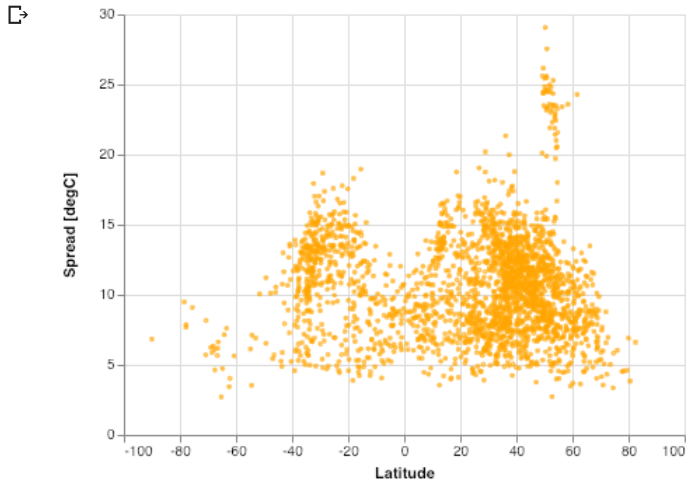






The map of average 1981-2018 temperature spreads by location show a few interesting features. First, the stations have poor coverage in the Amazon and much of Africa, which could bias results and limits applicability in those regions. Second, while there does not appear to be clear trends related to latitude or particular continents, it does appear that locations on islands and near coasts tend to have smaller temperature spreads. This makes sense given the ocean's high heat capacity. I do not end up incorporating a predictor for this, however. Last, and this will become evident later, in north-central Europe there is a region of anomalously high temperature spreads (30degC swings within a single month?).

```
## Elevation has some interesting nonlinearity
alt.Chart(station_attributes).mark_point(size=3, color='orange').encode(
  x=alt.X('lat', axis=alt.Axis(title="Latitude")),
  y=alt.Y('avg_spread', axis=alt.Axis(title='Spread [degC]'))
)
```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

Latitude has an interesting, nonlinear relationship to temperature spread, with values near the equator seeing very low monthly spreads, increasing as you expand out to about 20deg N/S, and then dropping off towards the poles. Because of this nonlinearity, it would not make sense to include it in the regression as is, and I instead bin latitude into a categorical predictor, based off of the relationships I see above.

```
# Define aggregate fields
lower_box = 'q1(avg_spread):Q'
lower_whisker = 'min(avg_spread):Q'
upper_box = 'q3(avg_spread):Q'
upper_whisker = 'max(avg_spread):Q'

# Compose each layer individually
lower_plot = alt.Chart(station_attributes).mark_rule().encode(
  x=alt.X(lower_whisker, axis=alt.Axis(title="Spread [degC]")),
  x2=lower_box,
  y='lat_bin:O'
)

middle_plot = alt.Chart(station_attributes).mark_bar(size=5.0, color='orange').encode(
  x=lower_box,
  x2=upper_box,
  y='lat_bin:O'
)

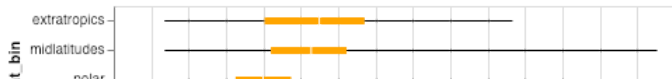
upper_plot = alt.Chart(station_attributes).mark_rule().encode(
  x=upper_whisker,
  x2=upper_box,
  y='lat_bin:O'
)

middle_tick = alt.Chart(station_attributes).mark_tick(
  color='white',
  size=10.0
).encode(
  x='median(avg_spread):Q',
  y='lat_bin:O',
)

lower_plot + middle_plot + upper_plot + middle_tick
```



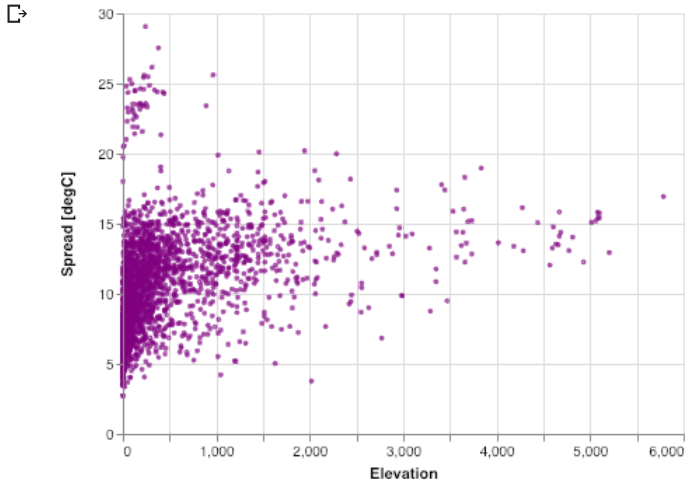




The new predictor related to latitude classifies latitude bands as tropics, extratropics, midlatitudes, and polar. From this plot we see that, as before, the spread is smallest near the poles and largest in the midlatitudes. It makes sense that the tropics see little temperature variability since they do not experience nearly as much of a seasonal cycle or variable weather systems.

[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

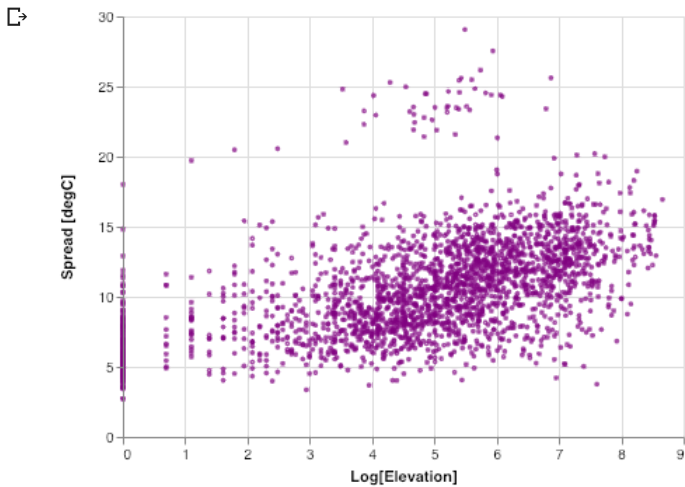
```
## Plot average spread vs. elevation
alt.Chart(station_attributes).mark_point(size=3, color='purple').encode(
  x=alt.X('elev', axis=alt.Axis(title="Elevation")),
  y=alt.Y('avg_spread', axis=alt.Axis(title='Spread [degC]'))
)
```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

Elevation appears to be related to temperature spreads, but it is not linear. Try some transformations:

```
## Try log elevation instead
alt.Chart(station_attributes).mark_point(size=3, color='purple').encode(
  x=alt.X('log_elev', axis=alt.Axis(title="Log[Elevation]")),
  y=alt.Y('avg_spread', axis=alt.Axis(title='Spread [degC]'))
)
```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

Taking a log-transform of elevation makes the relationship strongly linear. This implies temperatures are more variable at higher latitudes, which also matches the map from before in a way, where values near the coasts had low temperature variability. This relationship shown in the plot could be due to a combination of factors.

```
# Define aggregate fields
lower_box = 'q1(avg_spread):Q'
lower_whisker = 'min(avg_spread):Q'
upper_box = 'q3(avg_spread):Q'
upper_whisker = 'max(avg_spread):Q'

# Compose each layer individually
lower_plot = alt.Chart(station_attributes).mark_rule().encode(
  x=alt.X(lower_whisker, axis=alt.Axis(title="Spread [degC]")),
  x2=lower_box,
  y='elevation_bin:O'
)

middle_plot = alt.Chart(station_attributes).mark_bar(size=5.0, color='purple').encode(
  x=lower_box,
```

```

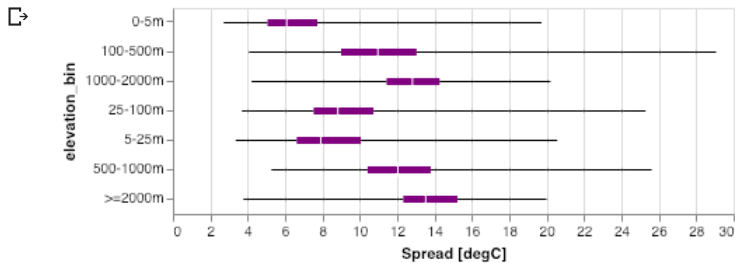
    x2=upper_box,
    y='elevation_bin:0'
)

upper_plot = alt.Chart(station_attributes).mark_rule().encode(
    x=upper_whisker,
    x2=upper_box,
    y='elevation_bin:0'
)

middle_tick = alt.Chart(station_attributes).mark_tick(
    color='white',
    size=10.0
).encode(
    x='median(avg_spread):Q',
    y='elevation_bin:0',
)

lower_plot + middle_plot + upper_plot + middle_tick

```



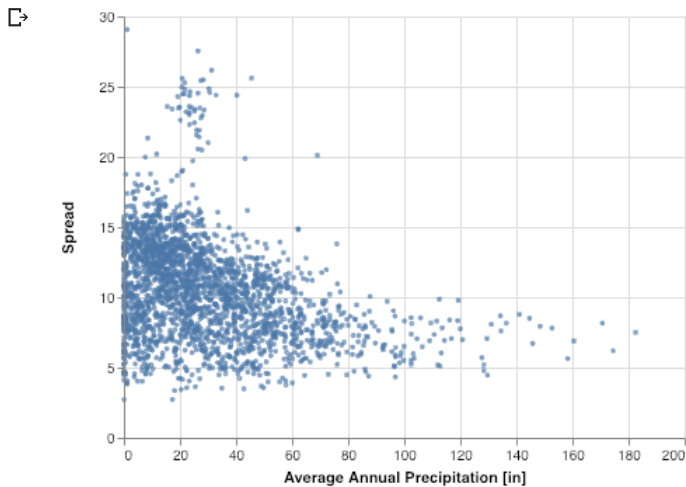
[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

Binning elevation is okay, but I will stick with the log-transform since the relationship is quite linear.

```

alt.Chart(station_attributes).mark_point(size=3).encode(
    x=alt.X('prcp_mean', axis=alt.Axis(title="Average Annual Precipitation [in]")),
    y=alt.Y('avg_spread', axis=alt.Axis(title='Spread [degC]'))
)

```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

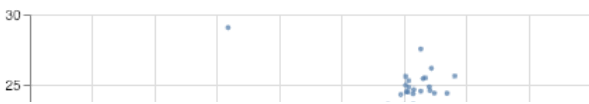
Precipitation may have a negative relationship with temperature spreads, but it is not particularly linear. Try some transformations.

```

alt.Chart(station_attributes).mark_point(size=3).encode(
    x=alt.X('log_precip', axis=alt.Axis(title="Log[Average Annual Precipitation]")),
    y=alt.Y('avg_spread', axis=alt.Axis(title='Spread [degC]'))
)

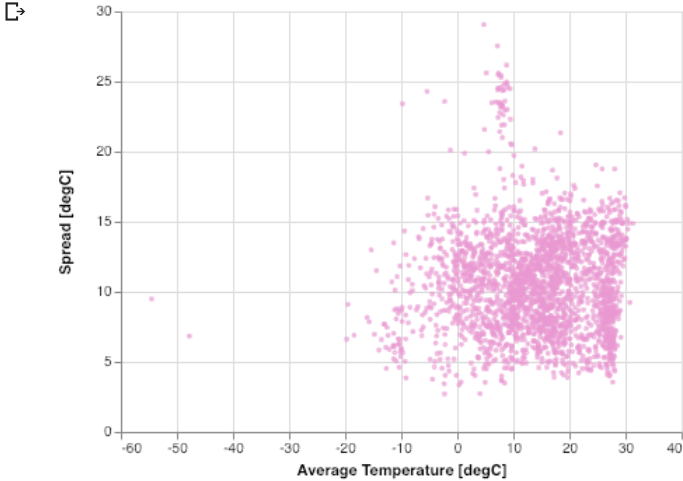
```

↗



The log-transform of precipitation doesn't look too bad. It is relatively linear aside from some high values of spread around log-precip=3 and the linear relationship does not extend to stations with little to no annual precipitation. A clever interaction term could address this in the regression, but it is not easily done via SQL/the BigQuery ML interface. Regardless, I chose to include log-precipitation as a predictor.

```
alt.Chart(station_attributes).mark_point(size=3, color='#EA98D2').encode(
  x=alt.X('tmean', axis=alt.Axis(title='Average Temperature [degC]')),
  y=alt.Y('avg_spread', axis=alt.Axis(title='Spread [degC]'))
)
```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

Temperature itself appears to have a slightly positive relationship with spread. The relationship actually appears to be slightly curved, suggesting the potential for a temperature<sup>2</sup> covariate. Spoiler, I include temperature and temperature<sup>2</sup> and they improve the model. Note that the temperatures used here are not used in the calculation of the response variable, tmax-tmin.

```
%%bigquery --project $project_id spread_season
SELECT id, AVG(spread) avg_spread, season
FROM (
  SELECT
    a.id,
    spread,
    a.year,
    a.month,
    CASE WHEN (a.month IN ('12','1','2') AND latitude>=0)
      OR (a.month IN ('6','7','8') AND latitude<0) THEN 'winter'
    WHEN (a.month IN ('3','4','5') AND latitude>=0)
      OR (a.month IN ('9','10','11') AND latitude<0) THEN 'spring'
    WHEN (a.month IN ('6','7','8') AND latitude>=0)
      OR (a.month IN ('12','1','2') AND latitude<0) THEN 'summer'
    WHEN (a.month IN ('9','10','11') AND latitude>=0)
      OR (a.month IN ('3','4','5') AND latitude<0) THEN 'fall'
    END AS season
  #-----
  FROM
    ## Saved table (response variable, year, month, tavg)
    `project3.results_20181130` a,
    ## Attributes of the station (e.g. elevation)
    `bigquery-public-data.ghcn.m.ghcnm_tmax_stations` b,
    ## Average 2000-2018 annual precipitation
    (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
     FROM (
       SELECT stn, SUM(prcp) AS prcp_annual, year
       FROM `bigquery-public-data.noaa_gsod.gsod200*`
       WHERE prcp != 99.99 AND year != '2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
       GROUP BY year,stn
     )
    GROUP BY stn
  ) c,
  ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
  (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
   FROM `bigquery-public-data.noaa_gsod.gsod200*`
   GROUP BY stn
  ) d
  #-----
  WHERE
    a.id=b.id AND grelev IS NOT NULL
    # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
    AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
    AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
    AND c.stn = d.stn
)
GROUP BY id, season
```

spread\_season.head(5)

```

id avg_spread season
0 42572797000 8.768242 fall
1 42572326000 12.088434 fall
2 50194843000 12.303333 fall
3 30886086000 10.855833 fall
4 16861976000 4.624444 fall

alt.data_transformers.enable('default', max_rows=None)
# Define aggregate fields
lower_box = 'q1(avg_spread):Q'
lower_whisker = 'min(avg_spread):Q'
upper_box = 'q3(avg_spread):Q'
upper_whisker = 'max(avg_spread):Q'

# Compose each layer individually
lower_plot = alt.Chart(spread_season).mark_rule().encode(
  x=alt.Y(lower_whisker, axis=alt.Axis(title="Spread [degC]")),
  x2=lower_box,
  y='season:O'
)

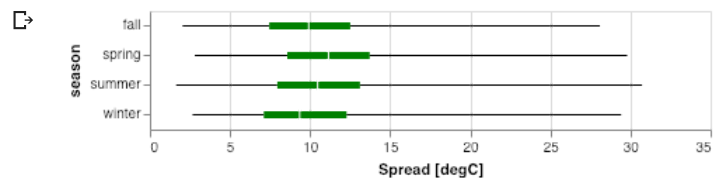
middle_plot = alt.Chart(spread_season).mark_bar(size=5.0, color='green').encode(
  x=lower_box,
  x2=upper_box,
  y='season:O'
)

upper_plot = alt.Chart(spread_season).mark_rule().encode(
  x=upper_whisker,
  x2=upper_box,
  y='season:O'
)

middle_tick = alt.Chart(spread_season).mark_tick(
  color='white',
  size=10.0
).encode(
  x='median(avg_spread):Q',
  y='season:O',
)

lower_plot + middle_plot + upper_plot + middle_tick

```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

The season of the year (adjusted for hemisphere) appears to have a moderate impact on temperature spread, with lower spreads in winter and higher variability in spring. This might make sense if you view the spring months as 'transition' months, where it's still trying to work things out between the relative stability of winter and summer. That logic doesn't necessarily transfer over to fall, however.

```

# Define aggregate fields
lower_box = 'q1(avg_spread):Q'
lower_whisker = 'min(avg_spread):Q'
upper_box = 'q3(avg_spread):Q'
upper_whisker = 'max(avg_spread):Q'

# Compose each layer individually
lower_plot = alt.Chart(station_attributes).mark_rule().encode(
  x=alt.Y(lower_whisker, axis=alt.Axis(title="Spread [degC]")),
  x2=lower_box,
  y='tornadoes:O'
)

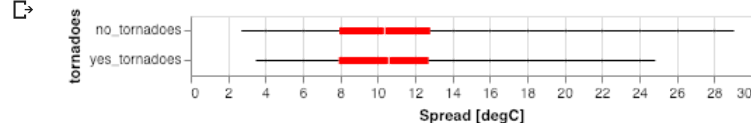
middle_plot = alt.Chart(station_attributes).mark_bar(size=5.0, color='red').encode(
  x=lower_box,
  x2=upper_box,
  y='tornadoes:O'
)

upper_plot = alt.Chart(station_attributes).mark_rule().encode(
  x=upper_whisker,
  x2=upper_box,
  y='tornadoes:O'
)

middle_tick = alt.Chart(station_attributes).mark_tick(
  color='white',
  size=10.0
).encode(
  x='median(avg_spread):Q',
  y='tornadoes:O',
)

lower_plot + middle_plot + upper_plot + middle_tick

```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

I mostly included the torndoes attribute because it caught my eye and seemed exciting. One could argue it is a proxy for whether the site experiences extreme weather and the meeting of warm and cold fronts, which would result in more temperature swings. Technically the metric is the presence of a 'tornado funnel cloud', and I classify each station as being tornado-possible or not based on whether it saw at least one funnel cloud between 2000 and 2018. Those annual thresholds were relatively arbitrary; I just wanted a sufficient sample size because tornadoes are relatively rare events. That being said, the median spread is higher for stations that have seen at least one funnel cloud, and though the difference is very small, I include it as a predictor.

```
%%bigquery --project $project_id spread_year
SELECT id, AVG(spread) avg_spread, CAST(year AS STRING) yr
FROM (
  SELECT
    a.id,
    spread,
    a.year
  #-----
  FROM
    ## Saved table (response variable, year, month, tavg)
    `project3.results_20181130` a,
    ## Attributes of the station (e.g. elevation)
    `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
    ## Average 2000-2018 annual precipitation
    (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
     FROM (
       SELECT stn, SUM(prcp) AS prcp_annual, year
       FROM `bigquery-public-data.noaa_gsod.gsod20*`
       WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
       GROUP BY year,stn
     )
     GROUP BY stn
    ) c,
    ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
    (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
     FROM `bigquery-public-data.noaa_gsod.gsod200*`
     GROUP BY stn
    ) d
  #-----
  WHERE
    a.id=b.id AND grelev IS NOT NULL
    # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
    AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
    AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
    AND c.stn = d.stn
)
GROUP BY id, CAST(year AS STRING)
```

```
spread_year.head(5)
```

	id	avg_spread	yr
0	20556018000	14.033333	1981
1	50194482000	14.518182	1981
2	20556571000	10.538333	1981
3	20551811000	13.491667	1981
4	42572654000	14.636667	1981

```
alt.data_transformers.enable('default', max_rows=None)
# Define aggregate fields
lower_box = 'q1(avg_spread):Q'
lower_whisker = 'min(avg_spread):Q'
upper_box = 'q3(avg_spread):Q'
upper_whisker = 'max(avg_spread):Q'

# Compose each layer individually
lower_plot = alt.Chart(spread_year).mark_rule().encode(
  y=alt.Y(lower_whisker, axis=alt.Axis(title="Spread [degC]")),
  y2=lower_box,
  x='yr:O'
)

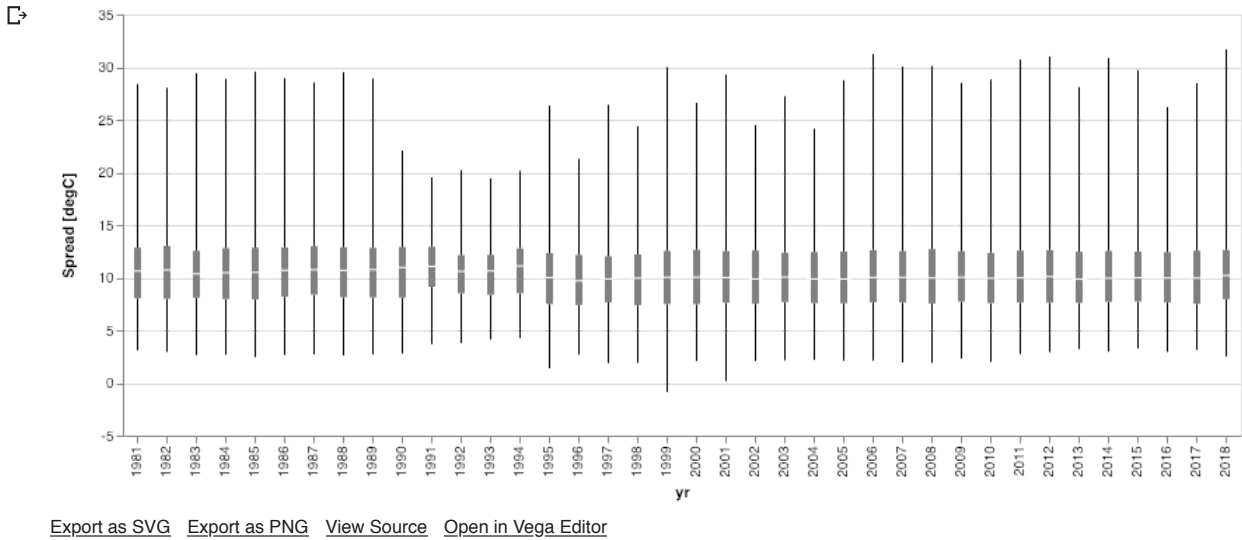
middle_plot = alt.Chart(spread_year).mark_bar(size=5.0, color='grey').encode(
  y=lower_box,
  y2=upper_box,
  x='yr:O'
)

upper_plot = alt.Chart(spread_year).mark_rule().encode(
  y=upper_whisker,
  y2=upper_box,
  x='yr:O'
)

middle_tick = alt.Chart(spread_year).mark_tick(
```

```
color='white',
size=10.0
).encode(
    y='median(avg_spread):Q',
    x='yr:O',
)

lower_plot + middle_plot + upper_plot + middle_tick
```



One logical question is whether temperatures have been getting more variable globally over time, potentially due to climate change. For whatever reason, based on the plot above the data suggests a step change to lower variability beginning in 1995, but not a linear trend. I am a bit suspicious as to why this might occur, so given more time one could calculate trends at each station and plot them on a map to see what else may be going on (not a SQL-friendly task).

```
%%bigquery --project $project_id spread_year_bin
SELECT id, AVG(spread) avg_spread, year_bin
FROM (
    SELECT
        a.id,
        spread,
        CASE WHEN a.year < 1995 then 'pre-1995'
              WHEN a.year >=1995 then '1995-2018'
        END AS year_bin
    #-----
    FROM
        ## Saved table (response variable, year, month, tavg)
        `project3.results_20181130` a,
        ## Attributes of the station (e.g. elevation)
        `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
        ## Average 2000-2018 annual precipitation
        (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
        FROM (
            SELECT stn, SUM(prcp) AS prcp_annual, year
            FROM `bigquery-public-data.noaa_gsod.gsod20*`
            WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
            GROUP BY year,stn
        )
        GROUP BY stn
    ) c,
    ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
    (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
    FROM `bigquery-public-data.noaa_gsod.gsod200*`
    GROUP BY stn
    ) d
    #-----
    WHERE
        a.id=b.id AND grelev IS NOT NULL
        # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
        AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
        AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
        AND c.stn = d.stn
)
GROUP BY id, year_bin
```

```
spread_year_bin.head(5)
```

	id	avg_spread	year_bin
0	63822113000	6.325000	pre-1995
1	50194680000	11.603488	pre-1995
2	20554027000	13.862577	pre-1995
3	20556748000	11.831183	pre-1995
4	42572315000	12.607551	pre-1995

```
# Define aggregate fields
```

```

lower_box = 'q1(avg_spread):Q'
lower_whisker = 'min(avg_spread):Q'
upper_box = 'q3(avg_spread):Q'
upper_whisker = 'max(avg_spread):Q'

# Compose each layer individually
lower_plot = alt.Chart(spread_year_bin).mark_rule().encode(
    x=alt.Y(lower_whisker, axis=alt.Axis(title="Spread [degC]")),
    x2=lower_box,
    y='year_bin:O'
)

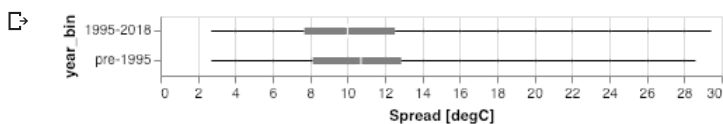
middle_plot = alt.Chart(spread_year_bin).mark_bar(size=5.0, color='grey').encode(
    x=lower_box,
    x2=upper_box,
    y='year_bin:O'
)

upper_plot = alt.Chart(spread_year_bin).mark_rule().encode(
    x=upper_whisker,
    x2=upper_box,
    y='year_bin:O'
)

middle_tick = alt.Chart(spread_year_bin).mark_tick(
    color='white',
    size=10.0
).encode(
    x='median(avg_spread):Q',
    y='year_bin:O'
)

lower_plot + middle_plot + upper_plot + middle_tick

```



[Export as SVG](#) [Export as PNG](#) [View Source](#) [Open in Vega Editor](#)

Based on the time series plot, instead of including 'year' as a predictor, either continuous or with the large sample size categorical, I instead create a new categorical predictor for whether or not the value occurred before 1995, and the plot above shows pre-1995 the spreads were slightly larger.

## ▼ Split data into training (70%), evaluation (20%), and test sets (10%)

Here I split the data (n=458,469) by subsetting by 'year', with the desired proportions accounted for in sampling the years pre-1995 and 1995-2018. For example, for the 14 years prior to 1995, I select 10 to be training, 2 to be evaluation and 2 to be test set, at random. I similarly break the 24 years from 1995-2018 into 17, 5, and 2. This leads to 27/38 years for training, 7/38 years for evaluation, and 4/38 years in the test set. My assumption here is that the sampling prior to 1995 is a 'random' sample, and the sampling post-1995 is a 'random' sample, which seems appropriate because aside for the shift beginning in 1995, year seems to be independent of average temperature spread.

Evaluation set: 1984,1989,1996,2000,2005,2010,2015 (19.8%)

Test set: 1986,1993,2002,2014 (9.6%)

```

%%bigquery --project $project_id

SELECT
    COUNT(a.id) AS n
FROM
    #-----
    ## Saved table (response variable, year, month, tavg)
    `project3.results_20181130` a,
    ## Attributes of the station (e.g. elevation)
    `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
    ## Average 2000-2018 annual precipitation
    (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
    FROM (
        SELECT stn, SUM(prcp) AS prcp_annual, year
        FROM `bigquery-public-data.noaa_gsod.gsod20*`
        WHERE prcp != 99.99 AND year != '2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
        GROUP BY year,stn
    )
    GROUP BY stn
) c,
    ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
    (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
    FROM `bigquery-public-data.noaa_gsod.gsod200*`
    GROUP BY stn
) d
    #-----
WHERE
    a.id=b.id AND grelev IS NOT NULL
    # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
    AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
    AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
    AND c.stn = d.stn

```





There are 458,469 individual values in the data set.

```
%%bigquery --project $project_id
SELECT n_eval/n*100 pct_eval
FROM(
SELECT
COUNT(a.id) AS n_eval

#-----
FROM
## Saved table (response variable, year, month, tavg)
`project3.results_20181130` a,
## Attributes of the station (e.g. elevation)
`bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
## Average 2000-2018 annual precipitation
(SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
FROM (
SELECT stn, SUM(prcp) AS prcp_annual, year
FROM `bigquery-public-data.noaa_gsod.gsod20`*
WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
GROUP BY year,stn
)
GROUP BY stn
) c,
## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200`*
GROUP BY stn
) d

#-----
WHERE
a.id=b.id AND grelev IS NOT NULL
# Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
AND c.stn = d.stn
AND a.year IN (1984,1989,1996,2000,2005,2010,2015)),

(SELECT
COUNT(a.id) AS n

#-----
FROM
## Saved table (response variable, year, month, tavg)
`project3.results_20181130` a,
## Attributes of the station (e.g. elevation)
`bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
## Average 2000-2018 annual precipitation
(SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
FROM (
SELECT stn, SUM(prcp) AS prcp_annual, year
FROM `bigquery-public-data.noaa_gsod.gsod20`*
WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
GROUP BY year,stn
)
GROUP BY stn
) c,
## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200`*
GROUP BY stn
) d

#-----
WHERE
a.id=b.id AND grelev IS NOT NULL
# Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
AND c.stn = d.stn)
```

↳

pct_eval
0 19.848452

The evaluation set consists of 19.8% of the observations.

```
%%bigquery --project $project_id
SELECT n_test/n*100 pct_test
FROM(
SELECT
COUNT(a.id) AS n_test

#-----
FROM
## Saved table (response variable, year, month, tavg)
`project3.results_20181130` a,
## Attributes of the station (e.g. elevation)
`bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
## Average 2000-2018 annual precipitation
(SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
FROM (
SELECT stn, SUM(prcp) AS prcp_annual, year
FROM `bigquery-public-data.noaa_gsod.gsod20`*
WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
GROUP BY year,stn
```

```

GROUP BY stn
) c,
## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200*`
GROUP BY stn
) d

#-----
WHERE
a.id=b.id AND grelev IS NOT NULL
# Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
AND c.stn = d.stn
AND a.year IN (1986,1993,2002,2014)),

(SELECT
COUNT(a.id) AS n

#-----
FROM
## Saved table (response variable, year, month, tavg)
`project3.results_20181130` a,
## Attributes of the station (e.g. elevation)
`bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
## Average 2000-2018 annual precipitation
(SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
FROM (
SELECT stn, SUM(prcp) AS prcp_annual, year
FROM `bigquery-public-data.noaa_gsod.gsod20*`
WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
GROUP BY year,stn
)
GROUP BY stn
) c,
## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200*`
GROUP BY stn
) d

#-----
WHERE
a.id=b.id AND grelev IS NOT NULL
# Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
AND c.stn = d.stn)

```

	pct_test
0	9.581891

The test set consists of 9.6% of the observations.

### Fit the linear regression model

Response variable, temperature spread, is continuous. 8 total predictors:

post

```

%%bigquery --project $project_id
#### Fit linear regression model ####

CREATE OR REPLACE MODEL `project3.final_model`
OPTIONS(model_type='linear_reg') AS
SELECT
  spread AS label, # Designate as response variable
  year_bin,
  season,
  lat_bin,
  log_elevation,
  log_prdp_mean_annual,
  tavg,
  tavg_sq,
  tornadoes

FROM (
  SELECT
    spread,
    a.year,
    CASE WHEN a.year < 1995 then 'pre-1995'
          WHEN a.year >=1995 then '1995-2018'
    END AS year_bin,
    # a.month,
    CASE WHEN (a.month IN ('12','1','2') AND latitude>=0)
      OR (a.month IN ('6','7','8') AND latitude<0) then 'winter'
      WHEN (a.month IN ('3','4','5') AND latitude>=0)
      OR (a.month IN ('9','10','11') AND latitude<0) then 'spring'
      WHEN (a.month IN ('6','7','8') AND latitude>=0)
      OR (a.month IN ('12','1','2') AND latitude<0) then 'summer'
      WHEN (a.month IN ('9','10','11') AND latitude>=0)
      OR (a.month IN ('3','4','5') AND latitude<0) then 'fall'
    END AS season,
    # longitude,

```

```

# latitude,
CASE WHEN ABS(latitude) BETWEEN 0 AND 30 then 'tropics'
  WHEN ABS(latitude) > 30 AND latitude <=60 then 'midlatitudes'
  WHEN ABS(latitude) > 60 AND latitude <=90 then 'polar'
END AS lat_bin,
tagv,
tagv*tagv AS tagv_sq,
# CASE WHEN grelev < 5 then '0-5m'
#   WHEN grelev >=5 AND grelev < 25 then '5-25m'
#   WHEN grelev >=25 AND grelev < 100 then '25-100m'
#   WHEN grelev >=100 AND grelev < 500 then '100-500m'
#   WHEN grelev >=500 AND grelev < 1000 then '500-1000m'
#   WHEN grelev >=100 AND grelev < 2000 then '1000-2000m'
#   WHEN grelev >=2000 then '>=2000m'
# END AS elevation_bin,
LOG(CASE WHEN grelev !=0 THEN grelev ELSE 1 END) AS log_elevation, # Tweak to adjust for log(0)
# prcp_mean_annual,
LOG(CASE WHEN prcp_mean_annual !=0 THEN prcp_mean_annual ELSE 1 END) AS log_prcp_mean_annual, # Tweak to adjust for log(0)
tornadoes

#-----
FROM
## Saved table (response variable, year, month, tagv)
`project3.results_20181130` a,
## Attributes of the station (e.g. elevation)
`bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
## Average 2000-2018 annual precipitation
(SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
FROM (
  SELECT stn, SUM(prcp) AS prcp_annual, year
  FROM `bigquery-public-data.noaa_gsod.gsod20`
  WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
  GROUP BY year,stn
)
GROUP BY stn
) c,
## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
(SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
FROM `bigquery-public-data.noaa_gsod.gsod200`
GROUP BY stn
) d

#-----
WHERE
a.id=b.id AND grelev IS NOT NULL
# Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
AND c.stn = d.stn
AND a.year IN (1981,1982,1983,1985,1987,1988,1990,1991,1992,1994,1995,1997,1998,1999,2001,2003,2004,2006,2007,2008,2009,2011,2012,2013,2
)

## You can ignore the error: Table has no schema: call 'client.get_table()'

```

Take a look at training stats:

```

%%bigquery --project $project_id
SELECT
*
FROM
ML.TRAINING_INFO(MODEL `project3.final_model`)

```

	training_run	iteration	loss	eval_loss	duration_ms	learning_rate
0	0	4	11.246570	11.178490	11603	1.6
1	0	3	11.284100	11.214655	11040	1.6
2	0	2	11.392618	11.328886	17672	0.8
3	0	1	12.353856	12.306719	11891	0.4
4	0	0	14.368915	14.349954	7077	0.2

These are not particularly interesting since I have a designated evaluation set, except it appears it is converging on a loss value as desired.

## ▼ Evaluate the model on the held out 20% set

```

%%bigquery --project $project_id
SELECT
*
FROM
ML.EVALUATE(MODEL `project3.final_model`, (
  SELECT
  spread AS label, # Designate as response variable
  year_bin,
  season,
  lat_bin,
  log_elevation,
  log_prcp_mean_annual,
  tagv,
  tagv_sq,
  tornadoes

```

```
FROM (SELECT
  spread,
  a.year,
  CASE WHEN a.year < 1995 then 'pre-1995'
        WHEN a.year >=1995 then '1995-2018'
  END AS year_bin,
  # a.month,
  CASE WHEN (a.month IN ('12','1','2') AND latitude>=0)
        OR (a.month IN ('6','7','8') AND latitude<0) then 'winter'
        WHEN (a.month IN ('3','4','5') AND latitude>=0)
        OR (a.month IN ('9','10','11') AND latitude<0) then 'spring'
        WHEN (a.month IN ('6','7','8') AND latitude>=0)
        OR (a.month IN ('12','1','2') AND latitude<0) then 'summer'
        WHEN (a.month IN ('9','10','11') AND latitude>=0)
        OR (a.month IN ('3','4','5') AND latitude<0) then 'fall'
  END AS season,
  # longitude,
  # latitude,
  CASE WHEN ABS(latitude) BETWEEN 0 AND 30 then 'tropics'
        WHEN ABS(latitude) > 30 AND latitude <=60 then 'midlatitudes'
        WHEN ABS(latitude) > 60 AND latitude <=90 then 'polar'
  END AS lat_bin,
  tavg,
  tavg*tavg AS tavg_sq,
  # CASE WHEN grelev < 5 then '0-5m'
  #   WHEN grelev >=5 AND grelev < 25 then '5-25m'
  #   WHEN grelev >=25 AND grelev < 100 then '25-100m'
  #   WHEN grelev >=100 AND grelev < 500 then '100-500m'
  #   WHEN grelev >=500 AND grelev < 1000 then '500-1000m'
  #   WHEN grelev >=100 AND grelev < 2000 then '1000-2000m'
  #   WHEN grelev >=2000 then '>=2000m'
  # END AS elevation_bin,
  LOG(CASE WHEN grelev !=0 THEN grelev ELSE 1 END) AS log_elevation, # Tweak to adjust for log(0)
  # prcp_mean_annual,
  LOG(CASE WHEN prcp_mean_annual !=0 THEN prcp_mean_annual ELSE 1 END) AS log_prcp_mean_annual, # Tweak to adjust for log(0)
  tornadoes
#-----
FROM
  ## Saved table (response variable, year, month, tavg)
  `project3.results_20181130` a,
  ## Attributes of the station (e.g. elevation)
  `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
  ## Average 2000-2018 annual precipitation
  (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
  FROM (
    SELECT stn, SUM(prcp) AS prcp_annual, year
    FROM `bigquery-public-data.noaa_gsod.gsod20*`
    WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
    GROUP BY year,stn
  )
  GROUP BY stn
  ) c,
  ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
  (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
  FROM `bigquery-public-data.noaa_gsod.gsod200*`
  GROUP BY stn
  ) d
#-----
WHERE
  a.id=b.id AND grelev IS NOT NULL
  # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
  AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
  AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
  AND c.stn = d.stn
  AND a.year IN (1984,1989,1996,2000,2005,2010,2015) ## THIS IS CRITICAL
)))
```

	mean_absolute_error	mean_squared_error	mean_squared_log_error	median_absolute_error	r2_score	explained_variance
0	2.476693	11.360356	0.08832	1.955533	0.305335	0.305336

Roughly 31% of the variability in temperature spread can be explained by the model on this evaluation set. That's not great, but it's better than I expected given we are looking at measurements across the globe. I also compared whether including the temperature<sup>2</sup> predictor improved estimates on this evaluation set, which it did very slightly, so I ultimately decided to include it in the model.

## ▼ Interpret model coefficients

```
%%bigquery --project $project_id
SELECT
  *
FROM
  ML.WEIGHTS(MODEL `project3.final_model`)
```







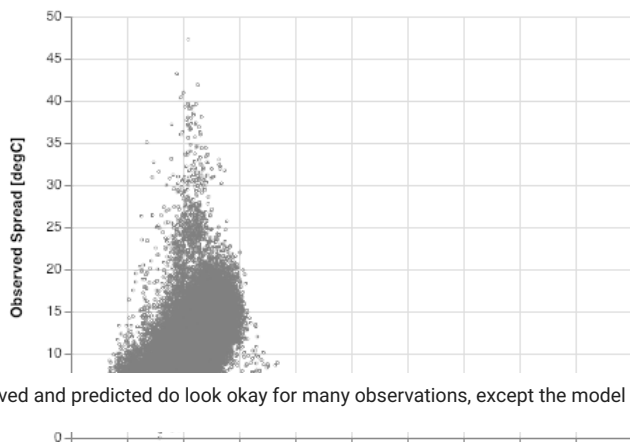
	mean_absolute_error	mean_squared_error	mean_squared_log_error	median_absolute_error	r2_score	explained_variance
0	2.467085	11.901389	0.086392	1.930975	0.300674	0.300965

```
%%bigquery --project $project_id test_preds
```

```
SELECT
  predicted_label, label
FROM
  ML.PREDICT(MODEL `project3.final_model`, (
    SELECT
      spread AS label, # Designate as response variable
      year_bin,
      season,
      lat_bin,
      log_elevation,
      log_prdp_mean_annual,
      tavg,
      tavg_sq,
      tornadoes
    FROM (SELECT
      spread,
      a.year,
      CASE WHEN a.year < 1995 then 'pre-1995'
            WHEN a.year >=1995 then '1995-2018'
      END AS year_bin,
      # a.month,
      CASE WHEN (a.month IN ('12','1','2') AND latitude>=0)
            OR (a.month IN ('6','7','8') AND latitude<0) then 'winter'
            WHEN (a.month IN ('3','4','5') AND latitude>=0)
            OR (a.month IN ('9','10','11') AND latitude<0) then 'spring'
            WHEN (a.month IN ('6','7','8') AND latitude>=0)
            OR (a.month IN ('12','1','2') AND latitude<0) then 'summer'
            WHEN (a.month IN ('9','10','11') AND latitude>=0)
            OR (a.month IN ('3','4','5') AND latitude<0) then 'fall'
      END AS season,
      # longitude,
      # latitude,
      CASE WHEN ABS(latitude) BETWEEN 0 AND 30 then 'tropics'
            WHEN ABS(latitude) > 30 AND latitude <=60 then 'midlatitudes'
            WHEN ABS(latitude) > 60 AND latitude <=90 then 'polar'
      END AS lat_bin,
      tavg,
      tavg*tavg AS tavg_sq,
      # CASE WHEN grelev < 5 then '0-5m'
      #       WHEN grelev >=5 AND grelev < 25 then '5-25m'
      #       WHEN grelev >=25 AND grelev < 100 then '25-100m'
      #       WHEN grelev >=100 AND grelev < 500 then '100-500m'
      #       WHEN grelev >=500 AND grelev < 1000 then '500-1000m'
      #       WHEN grelev >=1000 AND grelev < 2000 then '1000-2000m'
      #       WHEN grelev >=2000 then '>=2000m'
      # END AS elevation_bin,
      LOG(CASE WHEN grelev !=0 THEN grelev ELSE 1 END) AS log_elevation, # Tweak to adjust for log(0)
      # prcp_mean_annual,
      LOG(CASE WHEN prcp_mean_annual !=0 THEN prcp_mean_annual ELSE 1 END) AS log_prdp_mean_annual, # Tweak to adjust for log(0)
      tornadoes
    #-----
    FROM
      ## Saved table (response variable, year, month, tavg)
      `project3.results_20181130` a,
      ## Attributes of the station (e.g. elevation)
      `bigquery-public-data.ghcn_m.ghcnm_tmax_stations` b,
      ## Average 2000-2018 annual precipitation
      (SELECT stn, ROUND(AVG(prcp_annual),1) AS prcp_mean_annual
      FROM (
        SELECT stn, SUM(prcp) AS prcp_annual, year
        FROM `bigquery-public-data.noaa_gsod.gsod20*`
        WHERE prcp != 99.99 AND year !='2018' # Dont include precip totals for 2018 and don't include values flagged as missing data
        GROUP BY year,stn
      )
      GROUP BY stn
    ) c,
      ## Tornadoes (binary, at least 1 tornado funnel cloud in 2000-2018)
      (SELECT stn, CASE WHEN SUM(CAST(tornado_funnel_cloud AS INT64)) = 0 THEN 'no_tornadoes' ELSE 'yes_tornadoes' END tornadoes
      FROM `bigquery-public-data.noaa_gsod.gsod200*`
      GROUP BY stn
    ) d
  ) d
  #-----
WHERE
  a.id=b.id AND grelev IS NOT NULL
  # Below matches up the station names, where in 'a' dataset the name has a bit tacked on the beginning and end
  AND SUBSTR(CAST(a.id AS STRING),9) = '000' # Needs to have '000' at the end to be in the same dataset as gsod
  AND SUBSTR(CAST(a.id AS STRING),4,6) = c.stn # Remove first 3 digits (country/area info)
  AND c.stn = d.stn
  AND a.year IN (1986,1993,2002,2014) ## THIS IS CRITICAL
))
```

```
alt.Chart(test_preds).mark_point(size=.2, color='grey').encode(
  x=alt.X('predicted_label', axis=alt.Axis(title="Predicted Spread [degC]"), scale=alt.Scale(domain=(0,50))),
  y=alt.Y('label', axis=alt.Axis(title='Observed Spread [degC]'), scale=alt.Scale(domain=(0,50)))
)
```





Observed and predicted do look okay for many observations, except the model tends to vastly underestimate temperature spreads above 20degC.

## Conclusions

The model appears to explain roughly 30% of the variability in temperature spreads in held out test/evaluation sets, leaving roughly 70% of variability still unexplained. It appears to do well for temperature spreads less than 20degC but has trouble predicting the extreme. That is not great, but it is better than I expected given we are looking at measurements and proxies for physical mechanisms applied across the globe. If I restricted the analysis to just the U.S., or just to CA, with the same covariates I would expect to see even better performance. There was also the issue of the outliers in Europe which will affect the coefficient weights.

Model results suggest that temperature variability, at least my proxy for temperature variability increases with increasing elevation, temperature (nonlinear relationship), and log-precipitation. Variability is also low near the equator and highest in the midlatitudes, and variability is highest in spring months (taking account for hemisphere definition of 'spring'). The low variability in the tropics was the most clear and makes sense because the tropics do not really experience seasons as we know them, and weather is pretty consistent year-round. There was a curious feature where variability seemed to have a step change decrease in 1995, which I am skeptical of. There was an El Nino in 1982 and 1998 and Mt. Pinatubo in 1991 (big eruptions cool the atmosphere for 1-2 years) but nothing particularly interesting at a global scale in 1995.

With more time one could do a more in depth study of those outliers, and an easy step would be to just remove them from the analysis and not make predictions over that region. We already are limited in our ability to make predictions in the Amazon and most of Africa due to limited data availability. A separate model could be built for those outliers to tease out what is going on. Also regarding limited data variability, I did not filter for things like requiring each station to have at least 'x' years with data, which may be helpful for better estimating long-term trends. Stations coming on and offline will complicate trend estimation when the data is all pooled together as it is in this project. The behavior where temperature variability dropped beginning in 1995 could conceivably be due to many stations coming online in 1995 in areas with lower temperature variability (or the converse in terms of stations going offline).

Realistically you might expect the effects of, say, elevation on temperature spread to depend on other factors like its location on a continent and proximity to the jet stream. Those types higher order effects, where you expect the effect of a covariate on the response to depend on levels of a different covariate, could be tested with interaction terms. I also chose my study period relatively arbitrarily. One could extend the analysis back in time further to look more at decadal changes and increase sample sizes. Data quality deteriorates the farther back you go, of course.

Another possible next step would be to look at temperature spreads (tmax-tmin) within a single day as a response variable. That has even more direct interpretations and is more closer related to effects on ecosystems. It was my initial intention to do this, but the queries were not manageable for this project, so I opted for monthly tmax - monthly tmin instead.