Introduction to using PCA for model selection

```
library(rcf)
library(dplyr)
library(readr)
library(here)
library(ggplot2)
library(ggrepel)

my_directory <- here::here()
```

What is PCA and why use it?

Principal components analysis (PCA) is a statistical tool that can help to visualize the variance of more than 2 variables. In traditional model selection methods, we select models using just temperature and precipitation, but PCA allows us use as many variables as we would like to select models that best represent the climate futures we are interested in. If we want to select which models will best show changes in temperature, precipitation, relative humidity, growing degree days and freeze thaw cycles, PCA is the best tool to use. It essentially is able to condense the variance of all 5 of those variables into an x-y plot, and we can select which models show the most variability on that plot. A more in-depth explanation of PCA can be found here.

PCA in the rcf package

For more advanced users of the `rcf` package, models can be selected using PCA with a somewhat adjusted workflow. You can either use your own data and start at the `cf_pca()` function, or you can use the threshold values to calculate which models are most representative of the variables you are interested in.

If you would like to use your own data to calculate the PCA, you can skip down to the “PCA Calculation” section below.

If you want to use the data that is created from the threshold values, the first two steps in using PCA are exactly the same as using the quadrant method:

1. Download data using `rcf_data()`
2. Calculate threshold values using `calc_thresholds()`

To see how to do this, you can follow along with An Introduction to the Reproducible Climate Futures package(INSERT LINK).

```
# raw_data <- rcf_data(SiteID = "BAND",
#                      latitude = 35.75758546,
#                      longitude = -106.3054344,
#                      directory = my_directory,
#                      units = "imperial")

raw_data <- read_csv("https://irmadev.nps.gov/DataStore/DownloadFile/660685")
#> Rows: 2191480 Columns: 10
#> -- Column specification --------------------------------------------------------
#> Delimiter: ","
#> chr  (2): gcm, units
#> dbl  (7): yr, precip, tmin, tmax, tavg, rhmin, rhmax
```

```
#> dttm (1): date
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
thresholds <- calc_thresholds("BAND", data = raw_data, directory = my_directory, units = "imperial")
#> Adding missing grouping variables: `gcm`
#> Warning in calc_thresholds("BAND", data = raw_data, directory = my_directory, :
#> thresholds.csv generated successfully. DO NOT edit this csv in excel. File is
#> too large and data will be lost, causing errors in future calculations.
```

Once the first two steps are completed, we can move on to summarizing for PCA.

1. Summarize threshold values for PCA

The threshold data used for PCA are change values from past to future. Therefore, we calculate those values first, and can enter them into our `cf_pca()` function afterwards.

```
pca_summary <- summarize_for_pca("BAND", data = thresholds, future_year = 2040, directory =
         my_directory)
```

We can look at the data to see how changes compare between models and what changes will happen in the future, averaged over 30 years.

```
glimpse(pca_summary)
#> Rows: 40
#> Columns: 28
#> $ gcm                              <chr> "bcc-csm1-1-m.rcp45", "bcc-csm1-1-m.rc~
#> $ units                            <chr> "imperial", "imperial", "imperial", "i~
#> $ precip_change                    <dbl> -1.446844e-04, -2.085192e-05, -1.72290~
#> $ tmax_change                      <dbl> 3.279025, 4.144630, 3.680522, 4.412027~
#> $ tmin_change                      <dbl> 2.998992, 3.886409, 2.884021, 3.776245~
#> $ tavg_change                      <dbl> 3.139008, 4.015520, 3.282272, 4.094136~
#> $ rhmin_change                     <dbl> -1.55852479, -2.11348678, -2.54053649,~
#> $ rhmax_change                     <dbl> -1.3139573, -1.9414457, -2.3597907, -1~
#> $ heat_index_change                <dbl> 3.144134, 4.005301, 3.490723, 4.270961~
#> $ heat_index_ec_change             <dbl> 19.513333, 27.413333, 24.526667, 30.09~
#> $ heat_index_dan_change            <dbl> 0.00000000, 0.00000000, 0.00000000, 0.~
#> $ temp_over_95_pctl_change         <dbl> 38.17333, 45.60667, 42.99333, 46.36000~
#> $ temp_over_99_pctl_change         <dbl> 21.380000, 28.780000, 27.746667, 32.24~
#> $ temp_over_95_pctl_length_change <int> 62, 53, 38, 25, 33, 17, 34, 20, 36, 64~
#> $ temp_under_freeze_change         <dbl> -13.006667, -16.506667, -13.080000, -2~
#> $ temp_under_freeze_length_change <int> -10, -20, -13, -16, -13, -20, -11, -14~
#> $ temp_under_5_pctl_change         <dbl> -9.900000, -14.633333, -9.980000, -13.~
#> $ no_precip_change                 <dbl> 4.126667, 3.960000, 7.860000, 3.460000~
#> $ no_precip_length_change          <int> -7, -9, 1, -24, -6, -9, -18, -5, 27, -~
#> $ precip_95_pctl_change            <dbl> 0.30666667, -0.06000000, 0.26666667, 0~
#> $ precip_99_pctl_change            <dbl> 0.00000000, 0.16666667, 0.20000000, 0.~
#> $ precip_moderate_change           <dbl> 0.14666667, 0.24666667, 0.26666667, 0.~
#> $ precip_heavy_change              <dbl> 0.00000000, 0.00000000, 0.00000000, 0.~
#> $ freeze_thaw_change               <dbl> -14.73333, -20.53333, -11.27333, -13.4~
#> $ gdd_change                       <dbl> 21.66667, 26.90000, 20.98000, 27.34667~
#> $ frost_change                     <dbl> 3.7400000, 5.2066667, 2.7733333, 2.606~
#> $ grow_length_change               <dbl> 19.21698, 19.65878, 19.73613, 26.10803~
#> $ cf                               <chr> "Warm Dry", "Central", "Warm Dry", "Ho~
```

2. Calculate PCA

If you are using your own data, the only column that you need to have (other than the variables you are using for PCA) is a column labeled `gcm` with the GCMs you want to select from. Looking at more models will guarantee that you are selecting the models that are most representative of your chosen variables. As a reference, the MACA data that is used in this package has 20 models, each with RCPs 4.5 and 8.5, meaning that the PCA selects the 2 or 4 most representative models out of 40 options. It is important that the variables you are choosing match the column names of those variables, otherwise the function will throw an error.

If you are continuing from the workflow above, you can now choose which variables you are most interested in calculating your PCA from. There are two ways to choose variables from the threshold dataframe:

1. Select all threshold variables - If you want to look at all threshold variables, including temperature, precipitation and relative humidity, this function allows you to type in `"all_threshold"` into the `variables` argument to use all variables.
2. Select your own variables - If you want to choose fewer variables, there is an explanation of each variable in the `README` on the rcf landing page. To use these variables for PCA, the naming convention follows the naming convention in the “Column Name” column of the table in the “Data” section, with `_change` added to the end of the variable name. For example, if you want to look at growing season, precipitation and heat index dangerous, you would find that the column names for these variables are `grow_length`, `precip`, and `heat_index_dan` and you would enter `c("grow_length_change", "precip_change", "heat_index_dan_change")` into the `variables` argument. It is important that the variables you are choosing match the column names of those variables, otherwise the function will throw an error.

Now that you understand how to use the function, let’s see what we get from it!

```
pca_means <- cf_pca("BAND", data = pca_summary, variables = "all_threshold", directory = my_directory)
```

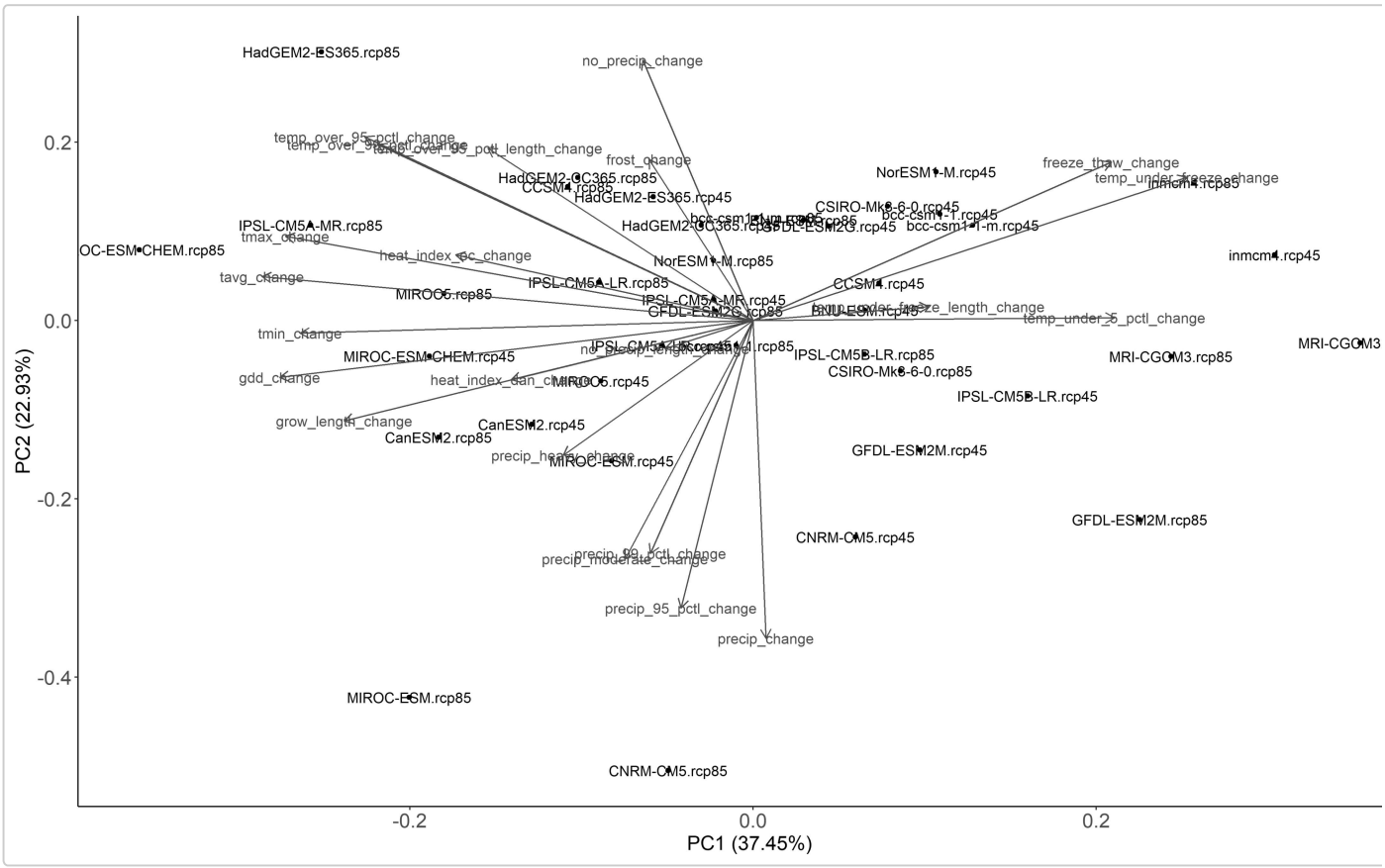
This function returns a dataframe and an image of the PCA plot and a dataframe named “BAND_future_means_pca.csv” with the models that have been selected for PCA. Any columns that have non-numeric values or NAs will be removed, as R cannot calculate PCA on columns that contain NAs.

First, let’s look at the dataframe.

```
glimpse(pca_means)
#> Rows: 40
#> Columns: 29
#> $ gcm                              <chr> "bcc-csm1-1-m.rcp45", "bcc-csm1-1-m.rc~
#> $ units                            <chr> "imperial", "imperial", "imperial", "i~
#> $ precip_change                    <dbl> -1.446844e-04, -2.085192e-05, -1.72290~
#> $ tmax_change                      <dbl> 3.279025, 4.144630, 3.680522, 4.412027~
#> $ tmin_change                      <dbl> 2.998992, 3.886409, 2.884021, 3.776245~
#> $ tavg_change                      <dbl> 3.139008, 4.015520, 3.282272, 4.094136~
#> $ rhmin_change                     <dbl> -1.55852479, -2.11348678, -2.54053649,~
#> $ rhmax_change                     <dbl> -1.3139573, -1.9414457, -2.3597907, -1~
#> $ heat_index_change                <dbl> 3.144134, 4.005301, 3.490723, 4.270961~
#> $ heat_index_ec_change             <dbl> 19.513333, 27.413333, 24.526667, 30.09~
#> $ heat_index_dan_change            <dbl> 0.00000000, 0.00000000, 0.00000000, 0.~
#> $ temp_over_95_pctl_change         <dbl> 38.17333, 45.60667, 42.99333, 46.36000~
#> $ temp_over_99_pctl_change         <dbl> 21.380000, 28.780000, 27.746667, 32.24~
#> $ temp_over_95_pctl_length_change <int> 62, 53, 38, 25, 33, 17, 34, 20, 36, 64~
#> $ temp_under_freeze_change         <dbl> -13.006667, -16.506667, -13.080000, -2~
#> $ temp_under_freeze_length_change <int> -10, -20, -13, -16, -13, -20, -11, -14~
#> $ temp_under_5_pctl_change         <dbl> -9.900000, -14.633333, -9.980000, -13.~
#> $ no_precip_change                 <dbl> 4.126667, 3.960000, 7.860000, 3.460000~
#> $ no_precip_length_change          <int> -7, -9, 1, -24, -6, -9, -18, -5, 27, -~
#> $ precip_95_pctl_change            <dbl> 0.30666667, -0.06000000, 0.26666667, 0~
```

```
#> $ precip_99_pctl_change          <dbl> 0.00000000, 0.16666667, 0.20000000, 0.~
#> $ precip_moderate_change         <dbl> 0.14666667, 0.24666667, 0.26666667, 0.~
#> $ precip_heavy_change            <dbl> 0.00000000, 0.00000000, 0.00000000, 0.~
#> $ freeze_thaw_change             <dbl> -14.73333, -20.53333, -11.27333, -13.4~
#> $ gdd_change                     <dbl> 21.66667, 26.90000, 20.98000, 27.34667~
#> $ frost_change                   <dbl> 3.7400000, 5.2066667, 2.7733333, 2.606~
#> $ grow_length_change             <dbl> 19.21698, 19.65878, 19.73613, 26.10803~
#> $ cf                             <chr> "Warm Dry", "Central", "Warm Dry", "Ho~
#> $ pca_type                       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
```

And the PCA plot.



This plot shows that 37.45% of the variance between all of these variables is represented by PC1 and 22.95% of the variance between these variables is represented by PC2. This indicates that these two components represent approximately 60% of the variance amongst all of these models.

Now that we have selected our models, we can go on to summarize the thresholds by PCA models using the `pca_thresholds()` function.

3. Summarize thresholds using PCA models

This function allows you to summarize your variables by month, season or year as well. You will need to enter two dataframes into this function:

1. `pca_data` is the resultant data from the `cf_pca()` function
2. `all_data` is the resultant data from the `calc_thresholds()` function

This function is only for the thresholds data that this package calculates. You can choose to select PCA models using whichever variables you want (including variables that are not included in the thresholds) and still be able to summarize threshold variables from the `calc_thresholds()` function by the selected PCA models. You cannot however, summarize a dataset that does not include all threshold variables, i.e. a dataset that has been modified after using the `calc_thresholds()` function.

```
threshold_summary <- pca_thresholds("BAND", pca_data = pca_means, all_data = thresholds, summarize_by =
        "year", directory = my_directory)
```

Now, let's look at the summary data:

```
glimpse(threshold_summary)
#> Rows: 328
#> Columns: 33
#> Groups: pca_type, yr, time [328]
#> $ pca_type                 <chr> "PC1 Max", "PC1 Max", "PC1 Max", "PC1 Max", "~
#> $ yr                       <dbl> 1950, 1951, 1952, 1953, 1954, 1955, 1956, 195~
#> $ time                     <fct> Historical, Historical, Historical, Historica~
#> $ gcm                      <chr> "MRI-CGCM3.rcp45", "MRI-CGCM3.rcp45", "MRI-CG~
#> $ cf                       <chr> "Warm Dry", "Warm Dry", "Warm Dry", "Warm Dry~
#> $ precip_yearly            <dbl> 13.14873, 21.94561, 12.78884, 11.69142, 21.53~
#> $ tmin                     <dbl> 36.15952, 39.45253, 38.22492, 37.65820, 37.30~
#> $ tmax                     <dbl> 64.35367, 66.20291, 66.42723, 66.86988, 64.40~
#> $ tavg                     <dbl> 50.25659, 52.82772, 52.32608, 52.26404, 50.85~
#> $ rhmin                    <dbl> 25.13386, 29.83231, 24.74927, 24.08987, 28.82~
#> $ rhmax                    <dbl> 70.86332, 78.63044, 70.22510, 67.38323, 77.65~
#> $ heat_index               <dbl> 61.09084, 63.46593, 63.32634, 63.65819, 61.46~
#> $ heat_index_ec            <int> 0, 0, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 0, 5, 0,~
#> $ heat_index_dan           <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ temp_over_95_pctl        <int> 11, 13, 19, 16, 9, 13, 22, 55, 1, 11, 3, 0, 7~
#> $ temp_over_99_pctl        <int> 0, 3, 1, 0, 0, 0, 2, 25, 0, 0, 0, 0, 0, 10, 0~
#> $ temp_over_95_pctl_length <int> 5, 4, 7, 6, 4, 8, 11, 14, 1, 6, 2, 0, 2, 14, ~
#> $ temp_under_freeze        <int> 154, 117, 133, 145, 142, 146, 129, 152, 156, ~
#> $ temp_under_freeze_length <int> 68, 27, 45, 62, 48, 35, 38, 65, 43, 33, 44, 4~
#> $ temp_under_5_pctl        <int> 44, 3, 11, 15, 20, 6, 11, 28, 37, 14, 35, 12,~
#> $ no_precip                <int> 293, 259, 284, 295, 253, 272, 297, 306, 279, ~
#> $ no_precip_length         <int> 59, 34, 36, 33, 26, 22, 34, 32, 24, 34, 28, 4~
#> $ precip_95_pctl           <int> 3, 9, 1, 3, 6, 7, 6, 3, 2, 1, 9, 2, 2, 3, 4, ~
#> $ precip_99_pctl           <int> 0, 0, 0, 0, 1, 4, 3, 0, 0, 0, 0, 0, 1, 0, 0, ~
#> $ precip_moderate          <int> 0, 0, 0, 0, 1, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ precip_heavy             <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
#> $ freeze_thaw              <int> 103, 68, 109, 111, 95, 110, 88, 104, 112, 110~
#> $ gdd                      <int> 239, 287, 254, 257, 258, 252, 274, 243, 226, ~
#> $ gdd_count                <int> 193, 209, 194, 178, 175, 207, 212, 199, 162, ~
#> $ not_gdd_count            <int> 45, 25, 44, 38, 27, 27, 31, 31, 43, 28, 36, 3~
#> $ frost                    <int> 28, 40, 21, 37, 35, 33, 39, 30, 17, 32, 30, 3~
#> $ grow_length              <dbl> 215, 280, 250, 251, 262, 255, 275, 251, 209, ~
#> $ units                    <chr> "imperial", "imperial", "imperial", "imperial~
```

4. Graphing

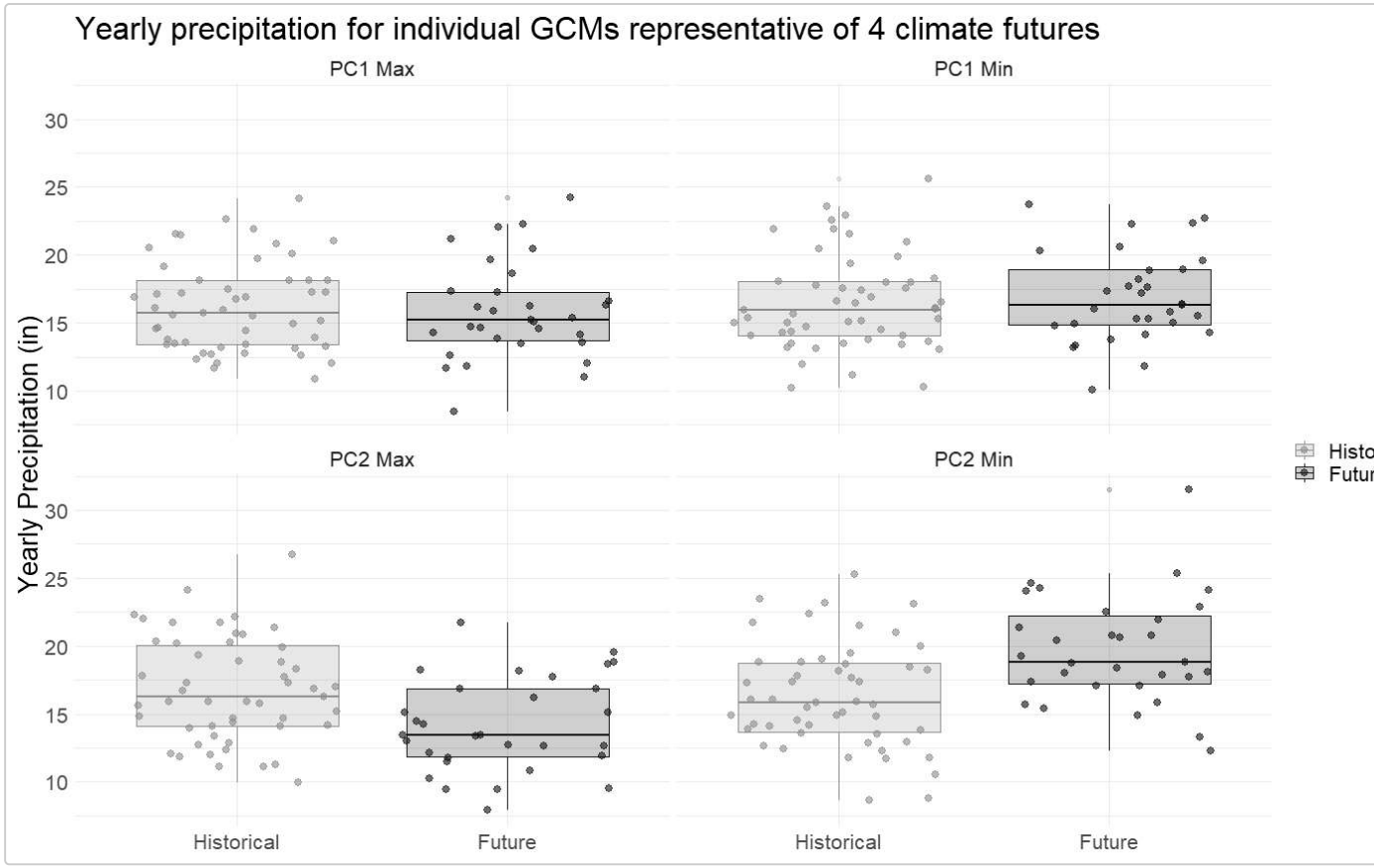
Because this is a dataframe, we can plot any of these variables using ggplot.

Let's look at the same graphs we looked at in An Introduction to the Reproducible Climate Futures package(INSERT LINK)

How does the precipitation change over time in the 4 climate futures?

```
ggplot(data = threshold_summary, aes(x = time, y = precip_yearly)) +
  geom_boxplot(aes(color = time,
                   fill = time),
               alpha = 0.2) +
```

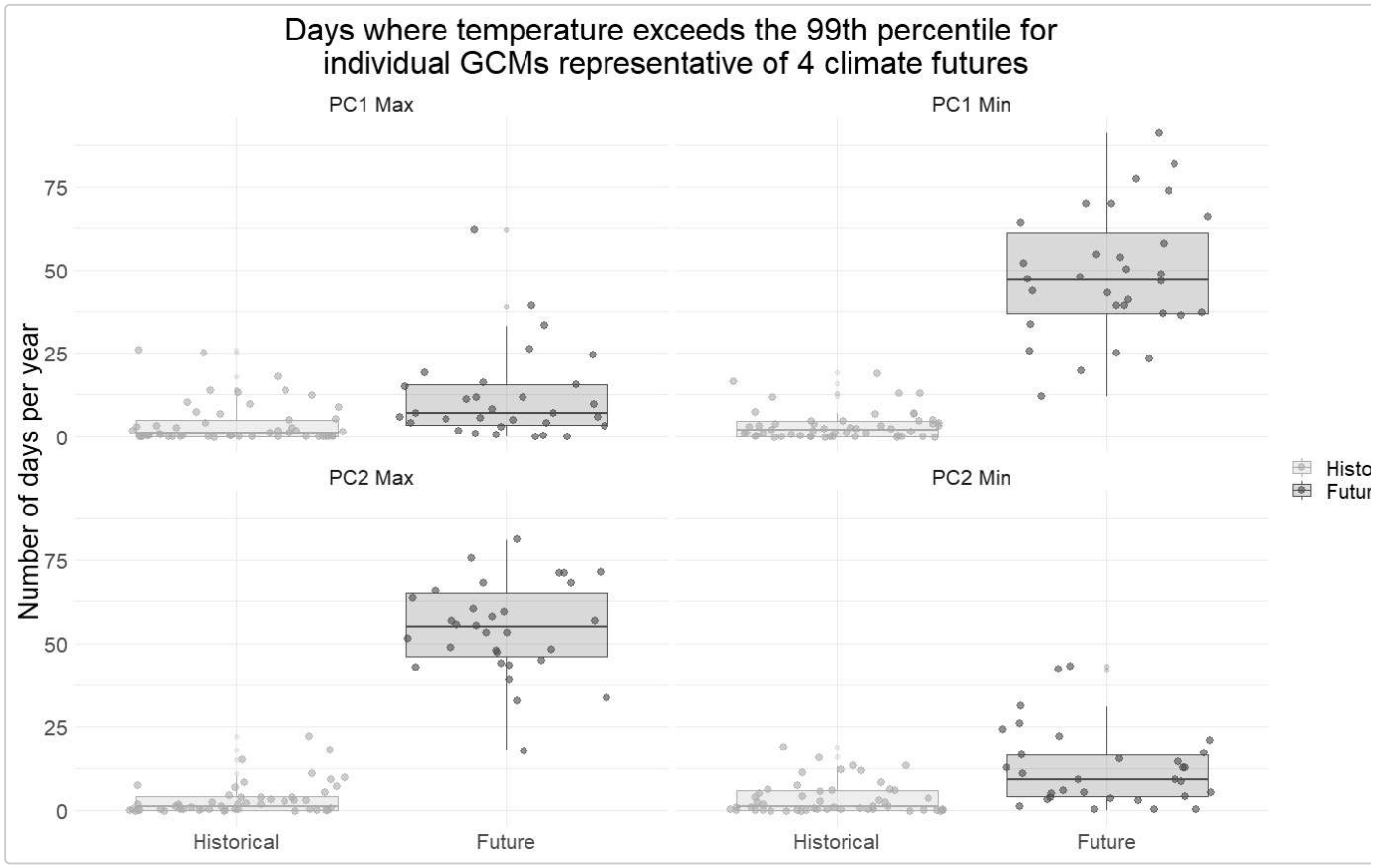
```
  geom_jitter(aes(color = time),
              size = 2.5,
              alpha =.6) +
  facet_wrap(~pca_type) +
  scale_color_manual(values = c("#8386CC", "#12045C")) +
  scale_fill_manual(values = c("#8386CC", "#12045C")) +
  labs(y = "Yearly Precipitation (in)",
       title = "Yearly precipitation for individual GCMs representative of 4 climate futures") +
  theme_minimal() +
  theme(text = element_text(size = 20),
        legend.title = element_blank(),
        axis.title.x = element_blank())
```



How do number of days that exceed the historical 99th percentile of heat compare in the past and future?

```
ggplot(data = threshold_summary, aes(x = time, y = temp_over_99_pctl)) +
  geom_boxplot(aes(color = time,
                   fill = time),
               alpha = 0.2) +
  geom_jitter(aes(color = time),
              size = 2.5,
              alpha =.6) +
  facet_wrap(~pca_type) +
  scale_color_manual(values = c("darksalmon", "#E10720")) +
  scale_fill_manual(values = c("darksalmon", "#E10720")) +
  labs(y = "Number of days per year",
       title = "Days where temperature exceeds the 99th percentile for\n individual GCMs representative
         of 4 climate futures") +
  theme_minimal() +
  theme(text = element_text(size = 20),
        legend.title = element_blank(),
```

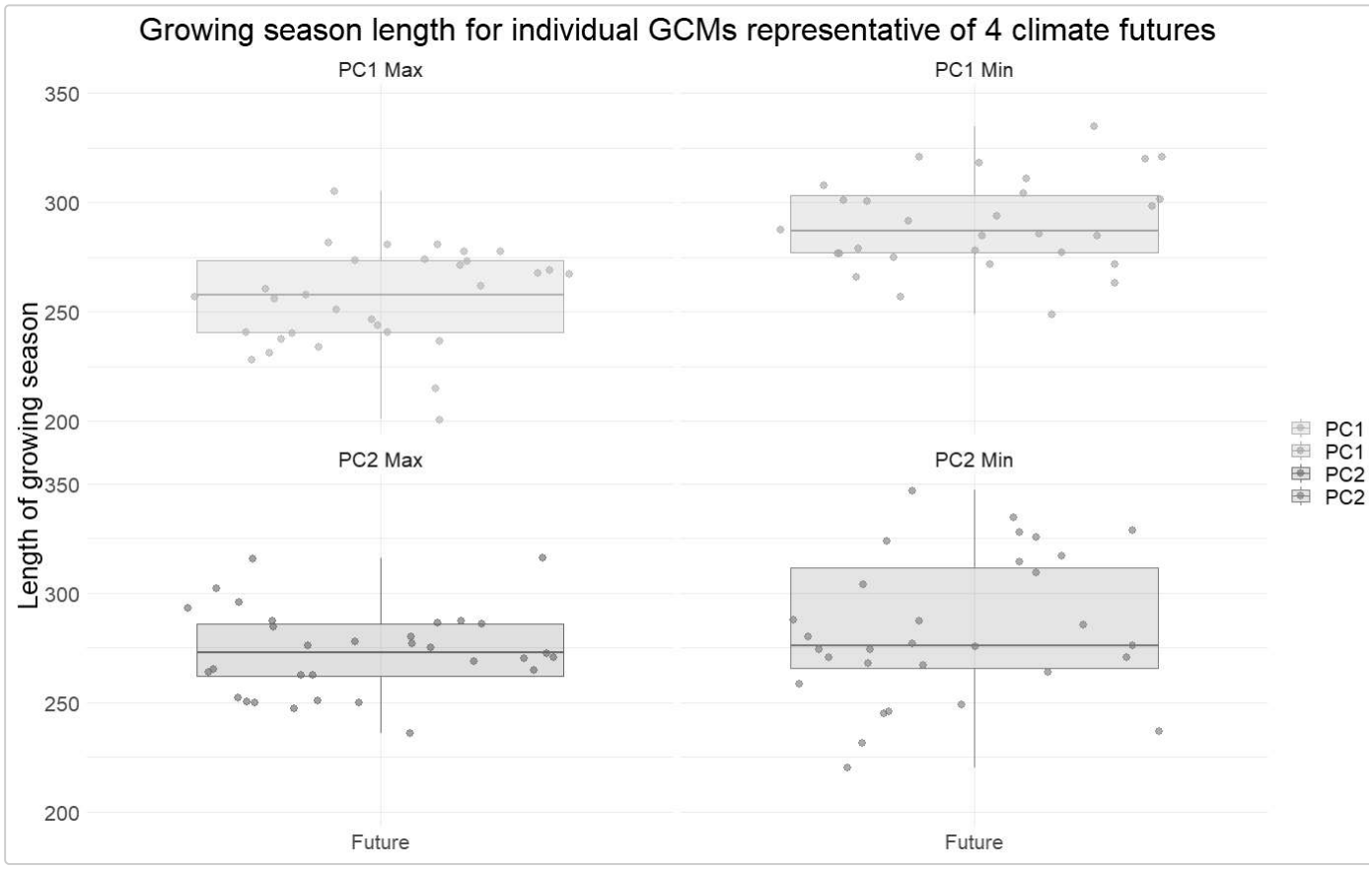
```
        axis.title.x = element_blank(),
        plot.title = element_text(hjust = 0.5))
```



What about the growing season length between models? How does that compare between the four climate futures?

```
threshold_summary_future <- threshold_summary %>%
filter(time %in% c("Future"))

ggplot(data = threshold_summary_future, aes(x = time, y = grow_length)) +
  geom_boxplot(aes(color = pca_type,
                   fill = pca_type),
               alpha = 0.2) +
  geom_jitter(aes(color = pca_type),
              size = 2.5,
              alpha =.6) +
  facet_wrap(~pca_type) +
  scale_color_manual(values = c("#8FD834", "#72CC50", "#019875", "#00AEAD")) +
  scale_fill_manual(values = c("#8FD834", "#72CC50", "#019875", "#00AEAD")) +
  labs(y = "Length of growing season",
       title = "Growing season length for individual GCMs representative of 4 climate futures") +
  theme_minimal() +
  theme(text = element_text(size = 20),
        legend.title = element_blank(),
        axis.title.x = element_blank(),
        plot.title = element_text(hjust = 0.5))
```


Growing season length for individual GCMs representative of 4 climate futures
PC1 Max
PC1 Min
PC2 Max
PC2 Min
Length of growing season
350
300
250
200
Future
PC1
PC1
PC2
PC2