Statistical Analysis: Predicting Trail Use

Dr. Meridith L. Bartley

2022-09-10

Contents

Section 1

Executive Summary

This report outlines a statistical analysis for predicting recreational trail use in the Bridger Mountains located near Bozeman, MT, USA. Headwaters Economics, in collaboration with U.S. Forest Service (USFS), [NAME] (GVSA), and Montana Fish Wildlife and Parks (FWP) have monitored trail use in the Bridger Mountains since 2021. In this report we present a modeling framework for modeling of nonlinear functional relationships between covariates and outcomes where the shape of the function itself varies between different grouping levels. We used the available data on 21 trails and trail subsections to fit several potential hierarchical generalized additive mixture models. These models varied in how much inter-trail variation was allowed when implementing a global soother in the model and in how we accounted for temporal autocorrelation.

To be added: Results from this analysis. But also, what do we want to highlight from this report for other stakeholders/interested groups?

Section 2

Introduction

We conduct a statistical analysis for multi-use recreational trails in the Bridger Mountains to estimate the volume and location of recreation. Headwaters Economics along with the U.S. Forest Service (USFS), [NAME?] (GVSA), and Montana Fish Wildlife and Parks (FWP) have deployed trail camera counters along a selection of recreational trails in the Bridger Mountains focused on the summer months of 2021.

The goals of this analysis is to (1) develop a predictive model of recreational trail use using data from trail counters, weather, trail characteristics, and novel data sources provided by Headwaters Economics and (2) apply the statistical model to demonstrate trade offs in predictive accuracy for different applications. The aim is to inform policy/methods adopted by land management agencies (e.g. U.S. Forest Service) and provide trend insights for local decision makers. Auxiliary data provided through partnerships with Strava and AllTrails was explored to examine the predictive capabilities of these data sources. We conducted analysis of pilot data for 21 trails subsections with trail cameras deployed. We used a generalized additive mixture model (`gamm`) approach which allows for estimation of trends as smooth, non-linear functions of time (and other covariates). We applied a framework of hierarchical generalized additive mixture model (HGAMM) (Pedersen et al., 2019) and various temporal autoregressive structures on the errors to address the spatial nesting and temporal correlation of these data, respectively. We present results for several models explored and examine the prediction estimates and residuals for both in- and out-of-sample trail subsections. We also provide recommendations on sampling effort (i.e. the spatial and temporal spread of trail camera deployment need to provide reliable predictions of trail use over time) for use both in the Bridgers and more generally.

An overview of the available data and covariates is presented in Section 3. An overview of the Generalized Additive Model framework, including models it builds upon and extensions within this framework, is covered in Section 4. These models are further explored and explained through an application (with R code and a primer on relevant diagnostic tools) to a single trail in Section 5. The main analysis of all trails (with monitoring via trail-use counters) within the Bridger Mountains is covered in Section 6. Trade-offs in prediction accuracy and various trail types is discussed in Section 7. Several short investigations covered during the development of this statistical analysis are included in Appendix Sections A and B.

This report is compiled using the following packages within the R statistical programming language (R Core Team, 2022): **bookdown** (Xie, 2020a), **rmarkdown** (Allaire et al., 2020), and **tinytex** (Xie, 2022). All associated code for analysis and this report may be found on the GitHub repository associated with this project: https://github.com/MLBartley/HE-TrailUse-R.

Section 3

Data Overview

For our analysis, we used trail use count data obtained from trail counters deployed along select multi-use recreational trails in the Bridger Mountains outside of Bozeman, Montana. These data will be used to create a predictive model for trail use over time for all trails in the Bridger Mountain range.

Data loading and wrangling is done with the **readxl** (Wickham and Bryan, 2019) and **tidyverse** (Wickham et al., 2019) packages.

Figures in this report are made with the **ggplot2** package (Wickham et al., 2020) and included using the **knitr** (Xie, 2020b) and **kableExtra** (Zhu, 2021) packages. The color palette is derived from the **pals** package (Wright, 2021). Interactive maps of trails and camera locations are created with the **mapview** package (Appelhans et al., 2022) and available in the HTML version of this report (not available in a PDF).

3.1 Sources of Data

We predict trail use from trail counters, weather, trail characteristics, and novel data sources provided by Headwaters Economics. The provided data sources are as follows:

1. Headwaters Economics Counter Data
2. Strava Data (aggregated trip counts)
3. AllTrails Data (daily search view count)
4. Weather covariates
5. Trail characteristics

3.1.1 Counter Data

Trail use camera counters (n = 33) were deployed at a subset of all trails (32) in the Bridger Mountains for 1 to 179 days. The counters record a use each time the beam is broken and were installed to minimize inaccurate counts from dogs or vegetation. Because this method measures total traffic, on a trail where use predominately is out-and-back the number of users will be approximately half of the total traffic. Some trails have multiple cameras deployed (e.g. Bridger Ridge). Trail subsections have been designated by Headwaters Economics that break up longer trails into shorter segments, each with its own camera(s). Metadata on trail camera hardware includes the counter owner. Metadata on trail camera location includes latitude and longitude. See Figure 3.1 for locations of trail counters in the Bridger Mountains.

In Figures Figure 3.3 and 3.4 we have time series plots for daily trail use counts separated by trail subsections (fill color indicates trail name). Most trail subsections have a single camera deployed, however a subset (Baldy to Bridger, Ross Pass to Sacagawea Peak, Sacagawea Pass, and Corbly Gulch) have two cameras. For

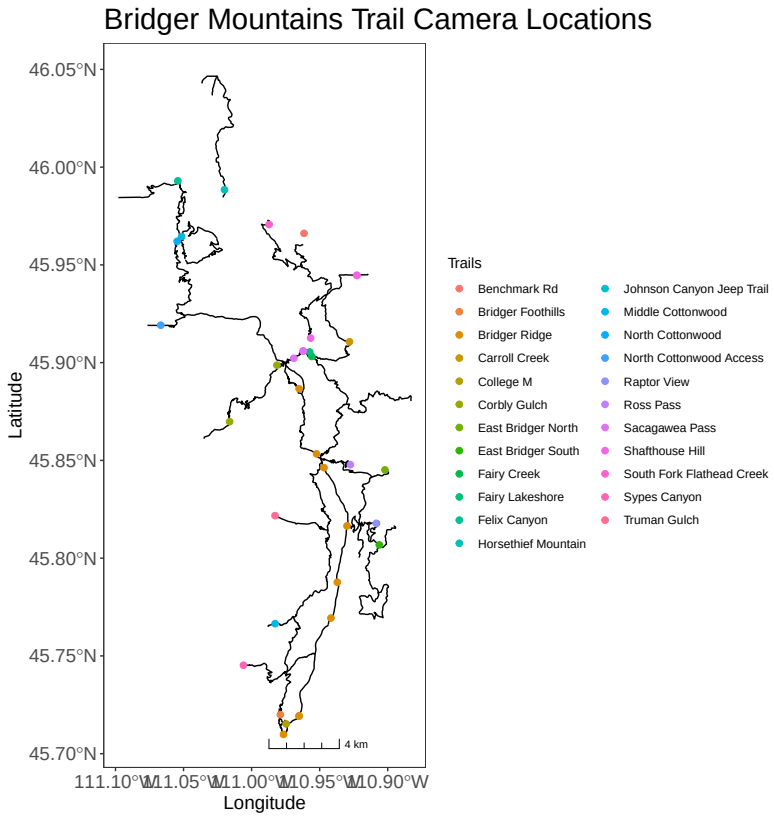


Figure 3.1: Locations Surveyed with infrared camera trail counters in Bridger Mountains

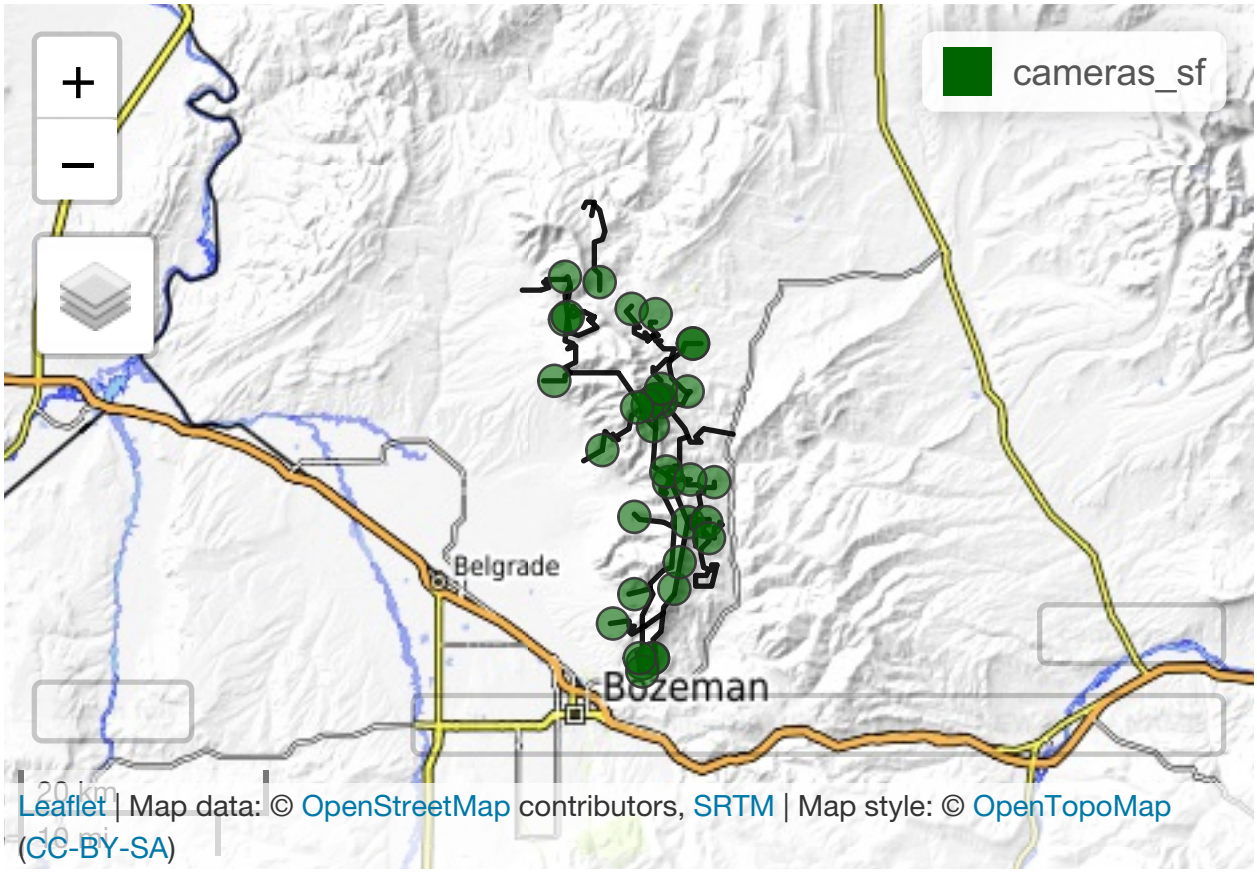


Figure 3.2: Interactive map of Bridger Mountain trails and camera counter locations. Available only in HTML format.

subsections with multiple cameras we have plotted the maximum count of trail uses between each camera per day. Several counters are placed in subsequent subsections along a single trail resulting in the capture of similar (i.e. non-independent) trail use information. For example, counter IDs 4, 5, 6, 7, and 9 are all located on Bridger Ridge and while the total counts for each counters are different (see Table 3.1) the time series plots show very similar patterns of use of time indicating non-independent counts.

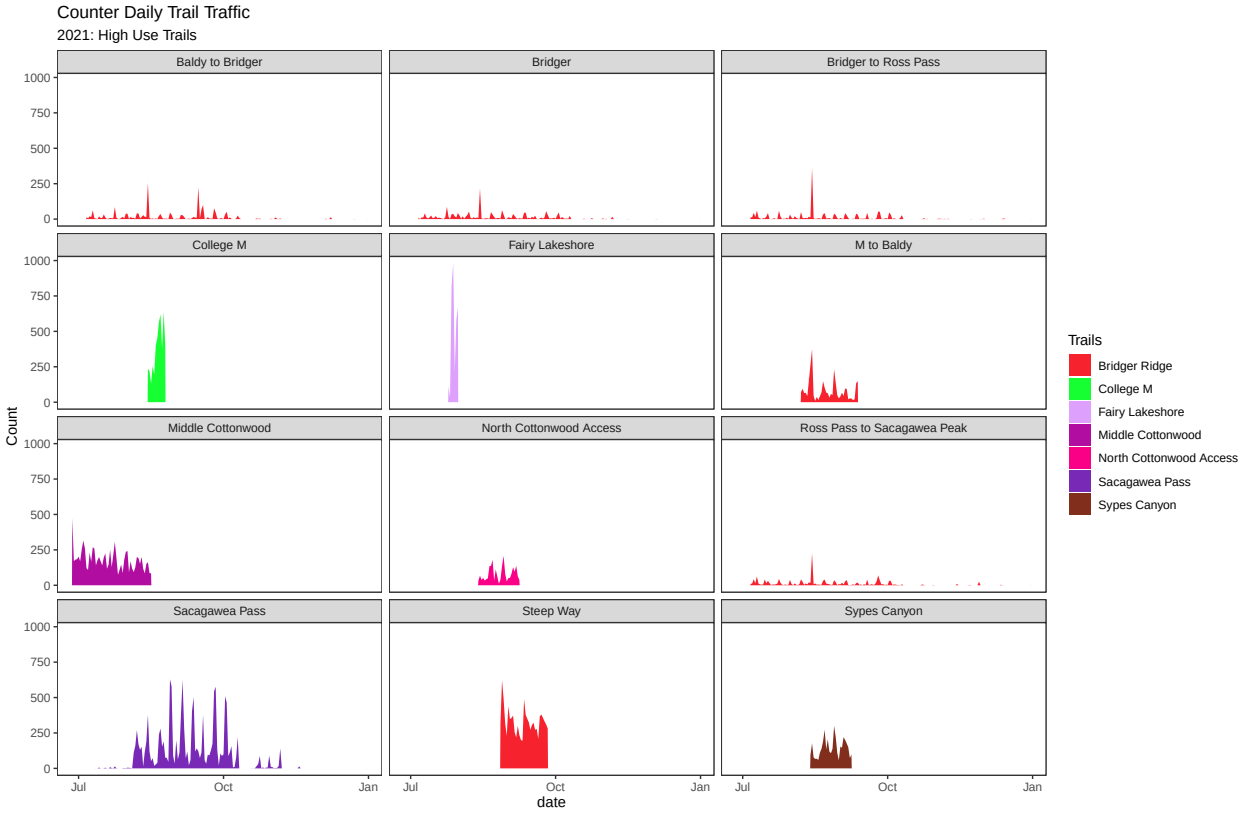


Figure 3.3: Timeseries plots of daily trail camera counts over time in the Bridger Mountains along high use trails. The following trails are not included in the analysis: Bridger, Bridger to Ross Pass, Fairy Lakeshore, M to Baldy, Ross Pass to Sacagawea Peak.

In Table 3.1 we provide a summary of trail use recorded for each counter deployed along trails in the Bridger Mountains. In addition to total counts for each counter, the deployment dates and duration of each trail counter camera (as determined by the first and last date of data provided) and average daily trail use over this deployment duration is reported.

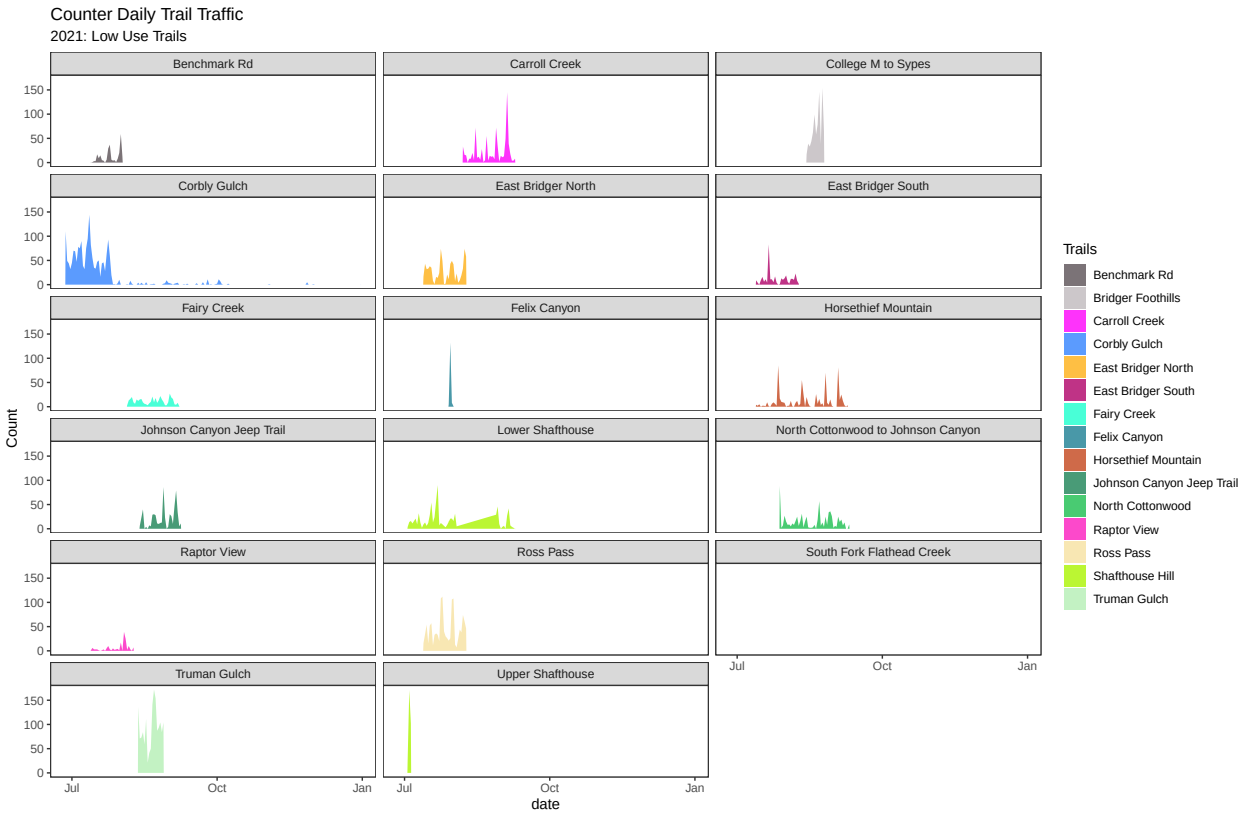


Figure 3.4: Timeseries plots of daily trail camera counts over time in the Bridger Mountains along low use trails. The following trails are not included in the analysis: Felix Canyon, South Fork Flathead Creek.

Table 3.1: Total Number of Trail Camera Counts (2021)

ID	Trail	Subsection	Count	Start	End	Deployment	Mean Count Per Day	Included in Analysis
1	Fairy Lakeshore	Fairy Lakeshore	3412	07-25	07-31	7 days	487.43	No
2	Fairy Creek	Fairy Creek	376	08-05	09-07	34 days	11.06	Yes
3	College M	College M	4493	08-14	08-25	12 days	374.42	Yes
4	Bridger Ridge	Baldy to Bridger	1261	07-06	12-31	179 days	7.04	Yes
5	Bridger Ridge	Baldy to Bridger	1844	07-06	12-31	179 days	10.30	Yes
6	Bridger Ridge	Bridger	1689	07-06	12-31	179 days	9.44	No
7	Bridger Ridge	Bridger to Ross Pass	1582	07-06	12-31	179 days	8.84	No
8	Bridger Ridge	M to Baldy	2635	08-07	09-12	37 days	71.22	No
9	Bridger Ridge	Ross Pass to Sacagawea Peak	1508	07-06	12-31	179 days	8.42	No
10	Bridger Ridge	Ross Pass to Sacagawea Peak	310	07-12	12-31	173 days	1.79	No
11	Bridger Ridge	Steep Way	10122	08-27	09-26	31 days	326.52	Yes
12	Sacagawea Pass	Sacagawea Pass	5192	08-05	09-09	36 days	144.22	Yes
13	Sacagawea Pass	Sacagawea Pass	9485	07-12	12-31	173 days	54.83	Yes
14	Horsethief Mountain	Horsethief Mountain	628	07-13	09-09	59 days	10.64	Yes
15	Carroll Creek	Carroll Creek	757	08-07	09-09	34 days	22.26	Yes
16	Felix Canyon	Felix Canyon	146	07-29	08-01	4 days	36.50	No
17	Raptor View	Raptor View	158	07-13	08-09	28 days	5.64	Yes
18	Sypes Canyon	Sypes Canyon	3868	08-13	09-08	27 days	143.26	Yes
19	Bridger Foothills	College M to Sypes	834	08-14	08-25	12 days	69.50	Yes
20	Truman Gulch	Truman Gulch	1585	08-12	08-28	17 days	93.24	Yes
21	East Bridger South	East Bridger South	321	07-13	08-09	28 days	11.46	Yes
22	East Bridger North	East Bridger North	780	07-13	08-09	28 days	27.86	Yes
23	Shafthouse Hill	Lower Shafthouse	687	07-03	09-09	69 days	9.96	Yes
24	Shafthouse Hill	Upper Shafthouse	303	07-03	07-05	3 days	101.00	No
25	South Fork Flathead Creek	South Fork Flathead Creek	91	07-03	07-03	1 days	91.00	No
26	Corbly Gulch	Corbly Gulch	1766	06-27	07-26	30 days	58.87	Yes
27	Corbly Gulch	Corbly Gulch	165	07-13	12-31	172 days	0.96	Yes
28	North Cottonwood	North Cottonwood to Johnson Canyon	662	07-28	09-10	45 days	14.71	Yes
29	North Cottonwood Access	North Cottonwood Access	2124	08-13	09-08	27 days	78.67	Yes
30	Ross Pass	Ross Pass	1246	07-13	08-09	28 days	44.50	Yes
31	Middle Cottonwood	Middle Cottonwood	9060	06-27	08-16	51 days	177.65	Yes
32	Johnson Canyon Jeep Trail	Johnson Canyon Jeep Trail	573	08-13	09-08	27 days	21.22	Yes
33	Benchmark Rd	Benchmark Rd	244	07-13	08-02	21 days	11.62	Yes

Due to low deployment times and possibly unreliable camera recordings, the following trails (numbers refer to counter IDs) are removed from this analysis:

- #1 - Fairy Lakeshore
- #16 - Felix Canyon
- #24 - Shafthouse Hill (Upper Shafthouse)
- #25 - South Fork Flathead Creek.

3.1.1.1 Hourly Data

For each trail counter camera deployed data is provided on a daily scale. Finer resolution data (i.e. hourly counts rather than daily) are available for 17 trails. Figures 3.5 and 3.6 shows trail use patterns with use hitting highest counts on the weekends, as expected.

The most likely use for these days is a quick look at how daily activity trends look (i.e. are there more hikers in mornings vs afternoon?). However since Strava data is not available at this resolution, it would not be easy to model this trend without a lot of data (i.e. counters deployed for longer amounts of time) to provide information on this trend in any model used.

Figure 3.5: Bridger Mountain trail traffic patterns by hour and day of week.

3.1.1.2 Strava Data

Strava count data are made available through Strava Metro. Data are binned (intervals of 5 with ceiling rounding) and aggregated on multiple scales (daily, monthly, annual). Counts are available as "total trips" and "total people". Total trips should always be larger than the total people count as people sometimes make

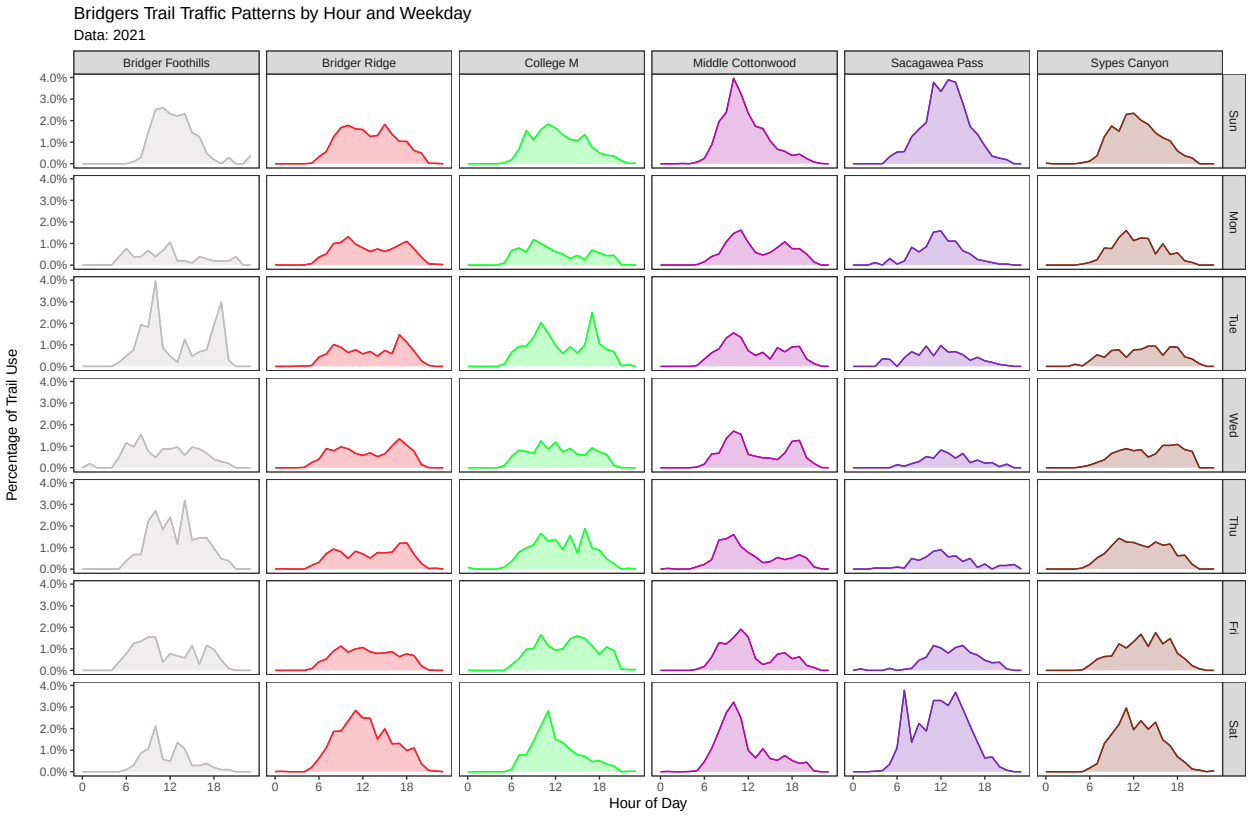


Figure 3.6: Bridger Mountain trail traffic patterns by hour and day of week.

Table 3.2: Total Number of Strava Counts (2021)

trailname	count	edges	Mean Count per Day	Included in Analysis
Benchmark Road	50	8	0.14	No
Bridger Foothills	61185	30	167.63	Yes
Bridger Ridge	56395	40	154.51	Yes
Carrol Creek	625	10	1.71	No
College M	27420	4	75.12	Yes
Corbly Gulch	12010	13	32.90	Yes
E Bridger North	4150	15	11.37	No
E Bridger South	70	4	0.19	No
Fairy Creek	2130	16	5.84	Yes
Fairy Lake	15	1	0.04	No
Fairy Lake Shortcut	305	1	0.84	No
Fairy Lakeshore	640	4	1.75	No
Felix Canyon Rd	885	7	2.42	No
Felix Canyon Trail	50	2	0.14	No
Flathead Pass Rd	755	18	2.07	No
Horsethief Mountain	30	3	0.08	Yes
Johnson Canyon Jeep Trail	160	11	0.44	Yes
M shortcut	5700	3	15.62	No
Middle Cottonwood	16135	8	44.21	Yes
New World Gulch	2140	6	5.86	No
North Cottonwood	4170	16	11.42	Yes
North Cottonwood Access	2525	4	6.92	Yes
Raptor View	570	4	1.56	Yes
Ross Pass	1600	4	4.38	Yes
S Fork Brackett Creek	280	3	0.77	No
S Fork Flathead Creek	5	1	0.01	No
Sacagawea Pass	2350	2	6.44	Yes
Shafthouse Hill	1640	9	4.49	Yes
Sypes Canyon	25575	11	70.07	Yes
Truman Gulch	7185	6	19.68	Yes
Upper Brackett Creek	430	6	1.18	No

multiple passes of a single trail (e.g. M laps). Strava trails are subdivided into "edges". Edge IDs (for the counter locations which often aligns with the trailhead edge) are available in the Counter data. Strava count data are available for the entire year of 2021 (not just summer monitoring as in the counter data). These data also include the overarching trail name and number (e.g. 511 - Bridger Foothills) that also correspond to the counter data provided by HE.

It is important to note that when considering Strava data at the Trail scale (rather than edge scale) you are propagating rounding errors for each segment forward (+_ 1-4 for each edge?). Similarly, aggregating the data in the daily data frame to a monthly timescale will likely not match the information provided in the monthly data frame.

Table 3.2 provides a summary of Strava data for the entire year. For each trail, the aggregated annual trail use count is provided as well as the number of Strava defined edges.

Figures 3.7 and 3.8 show time series plots for daily trail use counts (maximum number of trips over all edges in a trail) separated by trail (fill color indicates trail name).

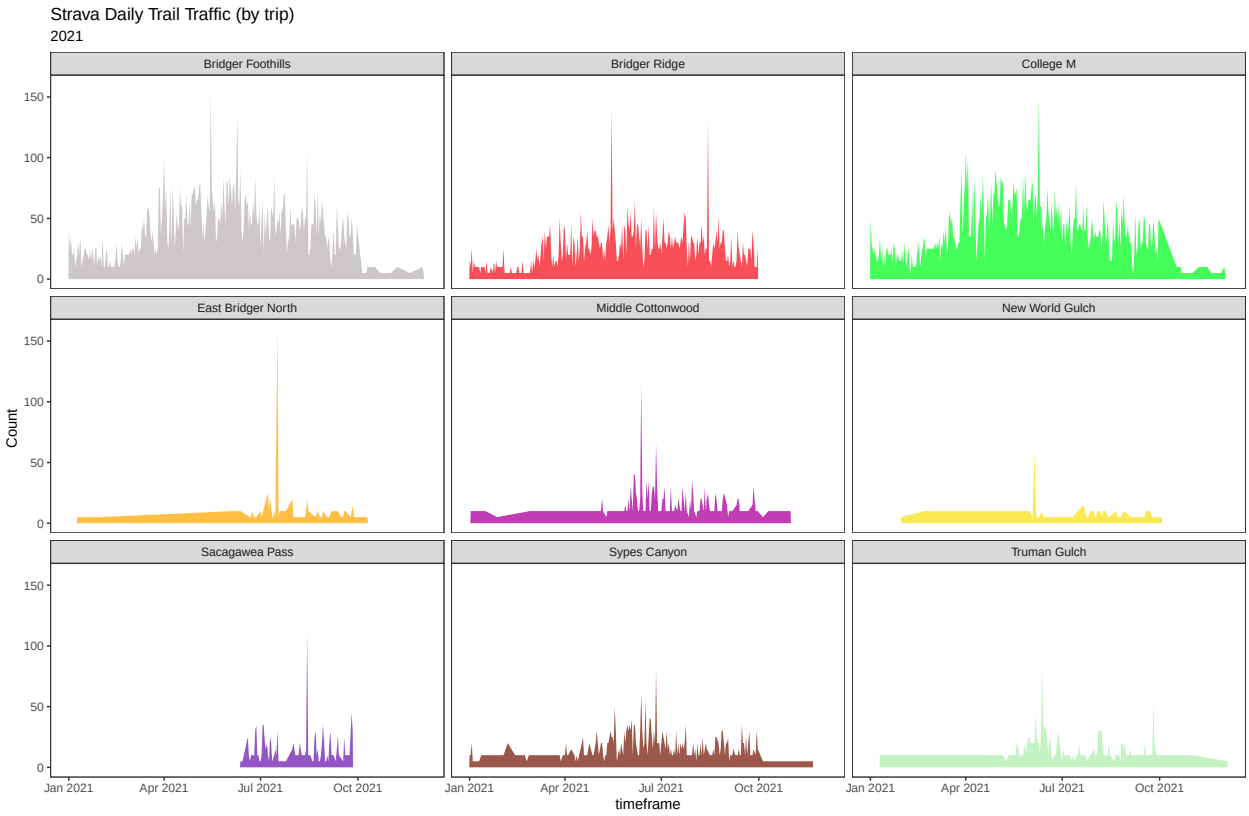


Figure 3.7: Timeseries plots of daily Strava trip counts over time in the Bridger Mountains along high use trails. The following trails are not included in the analysis: New World Gulch.

Figure 3.8: Timeseries plots of daily Strava trip counts over time in the Bridger Mountains along high use trails. The following trails are not included in the analysis: Fairy Lake Shortcut, Fairy Lakeshore, Felix Canyon Rd, M Shortcut, S Fork Brackett Creek.

3.1.1.3 AllTrails Data

AllTrails has provided number of daily searchs for each trail for 2020-2022. A seven day preceding moving average number of search terms was calculated for each trail. Figures 3.9 and 3.10 show time series plots for daily trail searches (7-day moving average) separated by trail (fill color indicates trail name).

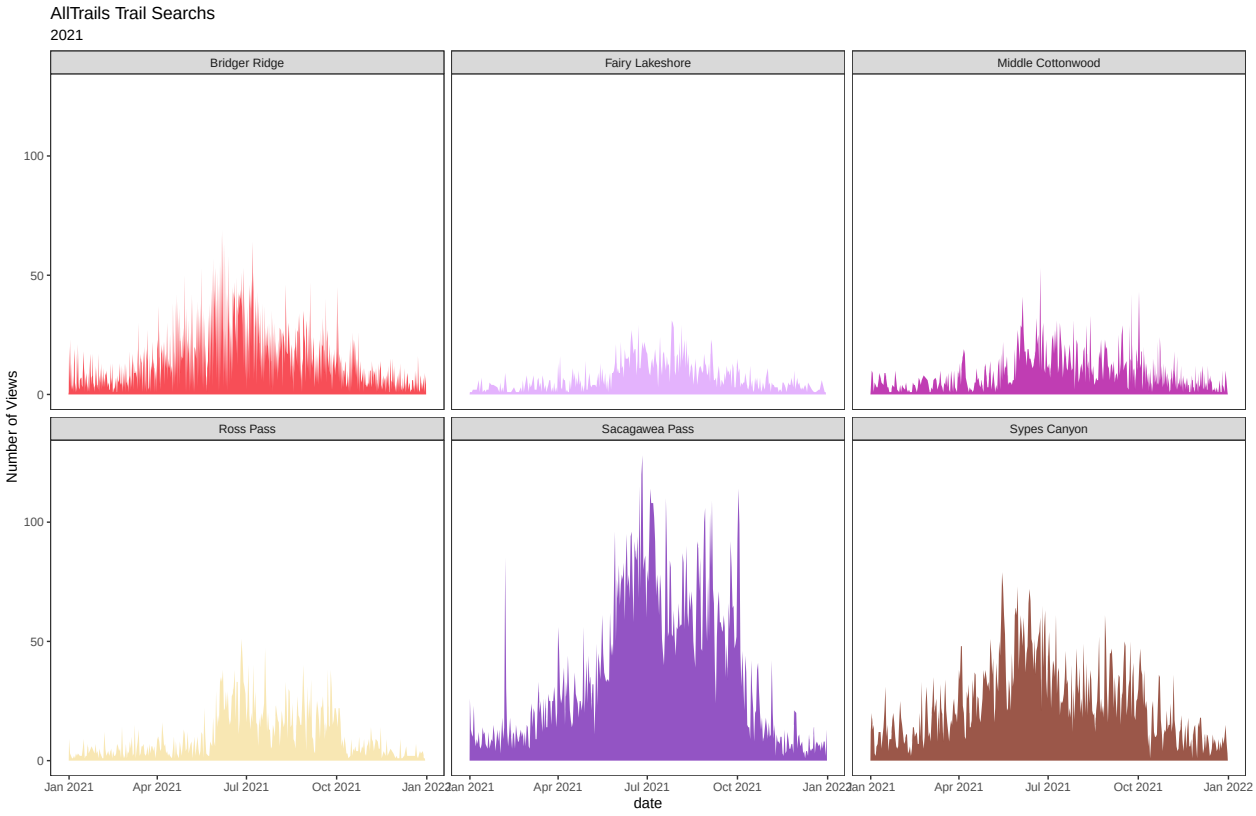


Figure 3.9: Timeseries plots of daily AllTrails trail searches as a moving average over time in the Bridger Mountains along high use trails.

3.1.2 Weather Covariates

The following weather covariates are available on a daily basis:

1. Precipitation (in.)
2. Temperature Max (degrees Fahrenheit)
3. Temperature Min (degrees Fahrenheit)
4. Mean Air Quality (AQI)
5. Mean PM_25 Concentration (micrograms per cubic meter)

These data do not vary spatially only temporally (i.e. the resolution is not fine enough to parse out different weather between trails on a given day). Figure 3.11 shows each covariate over time for 2021. Clear collinearity between several covariates (e.g. Min and Max Air Temperature) is apparent, and is considered when selecting covariates for inclusion in analysis.

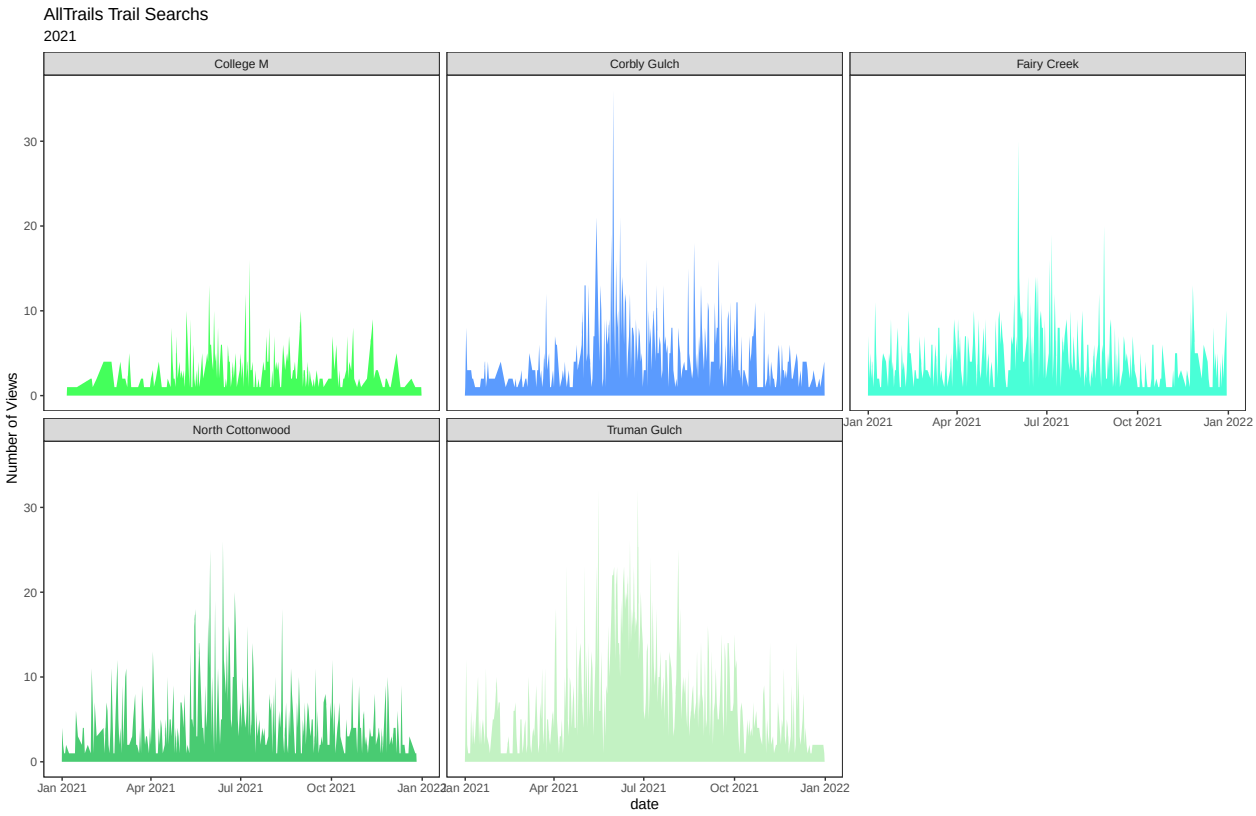


Figure 3.10: Timeseries plots of daily AllTrails trail searches as a moving average over time in the Bridger Mountains along low use trails.

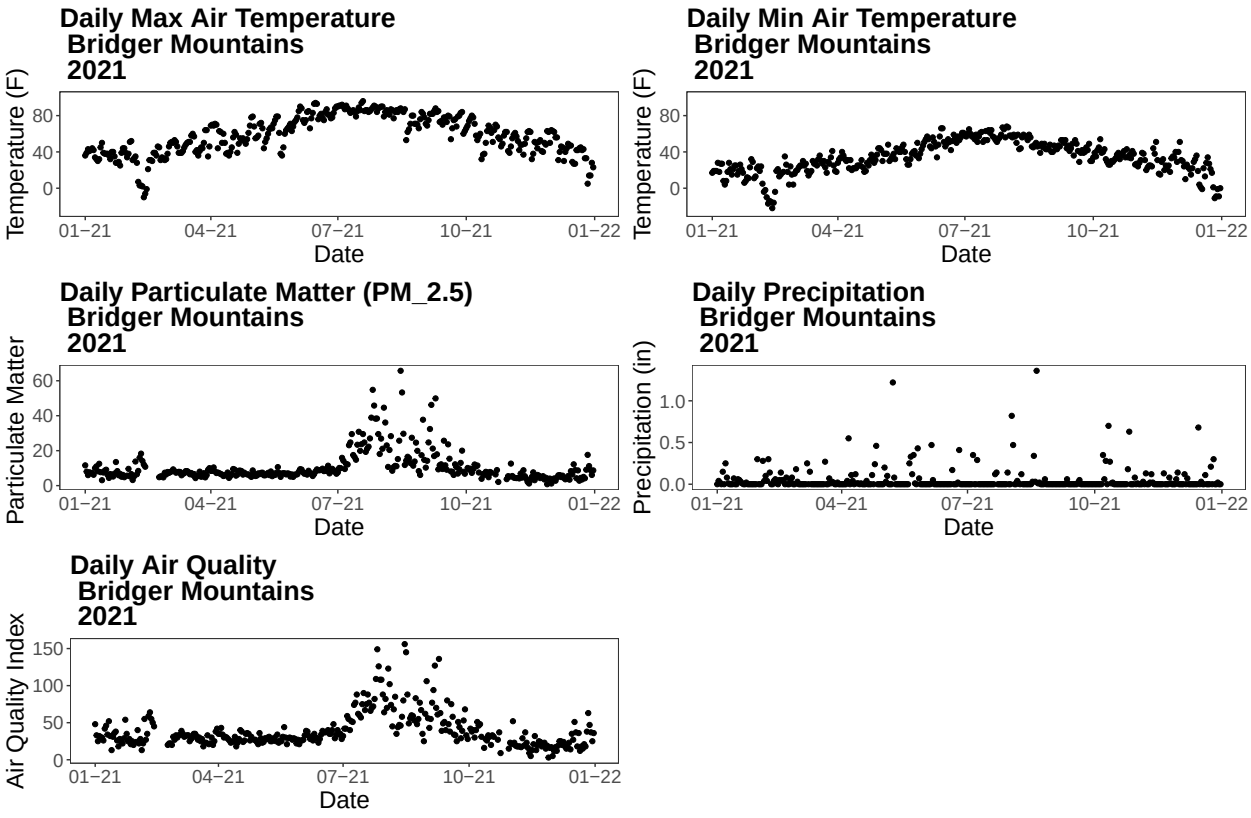


Figure 3.11: Weather covariate values over time.

3.1.3 Trail Characteristics

The following characteristics are provided at the trail level.

- Latitude/Longitude (of trailhead and camera location)
- Time/Distance from Bozeman, MT
- Parking lot size (as three-level factor)
- Description/Class (4-level factor of development)
- Motorized vehicle use (3-level factor)
- Indicators for use of the following:
  - Dirt bikes
  - ATVS
  - Hiking
  - Pack/Saddle
  - Bicycle

Table 3.3 provides an overview of these summaries.

Table 3.3: Bridger Mountain Trail Characteristics (2021)

Name	Subsection	Lot Size	Developed	Moterized Vehicle Use	Drive Time (mins)
Fairy Lakeshore	Fairy Lakeshore	L	Moderately Developed	Non-Motorized	57.16
Fairy Creek	Fairy Creek	L	Developed	Wheeled OHV 50" or <	46.39
Fairy Lake	Fairy Lake	L	Moderately Developed	Non-Motorized	57.16
College M	College M	L	Highly Developed	Non-Motorized	12.16
Bridger Ridge	Baldy to Bridger	L	Minimally Developed	Non-Motorized	12.16
Bridger Ridge	Bridger	L	Minimally Developed	Non-Motorized	31.70
Bridger Ridge	Bridger to Ross Pass	S	Minimally Developed	Non-Motorized	41.00
Bridger Ridge	M to Baldy	L	Minimally Developed	Non-Motorized	12.16
Bridger Ridge	Ross Pass to Sacagawea Peak	L	Minimally Developed	Non-Motorized	41.00
Bridger Ridge	Sacagawea Peak to Sacagawea Pass	L	Minimally Developed	Non-Motorized	57.16
Bridger Ridge	Steep Way	L	Moderately Developed	Non-Motorized	12.16
Sacagawea Pass	Sacagawea Pass	L	Developed	Non-Motorized	57.16
Horsethief Mountain	Horsethief Mountain	S	Developed	Non-Motorized	78.00
Carroll Creek	Carroll Creek	L	Highly Developed	Wheeled OHV 50" or <	46.39
Felix Canyon Trail	Felix Canyon Trail	S	Highly Developed	Wheeled OHV 50" or <	43.56
Raptor View	Raptor View	M	Developed	Non-Motorized	31.70
Sypes Canyon	Sypes Canyon	M	Developed	Non-Motorized	14.45
Bridger Foothills	Bostwick to Truman	L	Minimally Developed	Non-Motorized	25.74
Bridger Foothills	College M	L	Developed	Non-Motorized	12.16
Bridger Foothills	College M to Sypes	L	Developed	Non-Motorized	12.16
Bridger Foothills	Jones to Ross Pass	S	Minimally Developed	Non-Motorized	41.00
Bridger Foothills	Middle Cottonwood to Bostwick	M	Developed	Non-Motorized	17.78
Bridger Foothills	Ross Pass to Sacagawea Pass	S	Developed	Non-Motorized	41.00
Bridger Foothills	Sypes to Middle Cottonwood	M	Developed	Non-Motorized	14.45
Bridger Foothills	Truman to Jones	L	Minimally Developed	Non-Motorized	25.74
Truman Gulch	Truman Gulch	L	Developed	Dirt Bikes (seasonal)	25.74
East Bridger South	East Bridger South	L	Highly Developed	Non-Motorized	31.70
East Bridger North	East Bridger North	M	Developed	Non-Motorized	27.43
Shafthouse Hill	Lower Shafthouse	M	Developed	Non-Motorized	40.92
Shafthouse Hill	Upper Shafthouse	S	Developed	Non-Motorized	54.30
South Fork Flathead Creek	South Fork Flathead Creek	S	Highly Developed	Wheeled OHV 50" or <	49.88
Corbly Gulch	Corbly Gulch	S	Developed	Dirt Bikes (seasonal)	26.42
North Cottonwood	North Cottonwood to Johnson Canyon	L	Developed	Non-Motorized	39.33
North Cottonwood	North Cottonwood to Ridge	M	Developed	Non-Motorized	33.46
North Cottonwood Access	North Cottonwood Access	M	Developed	Non-Motorized	33.46
Ross Pass	Ross Pass	S	Developed	Dirt Bikes (seasonal)	41.00
Middle Cottonwood	Middle Cottonwood	M	Highly Developed	Dirt Bikes (seasonal)	17.78
Johnson Canyon Jeep Trail	Johnson Canyon Jeep Trail	L	Highly Developed	Wheeled OHV 50" or <	39.33
Felix Canyon Rd	Felix Canyon Rd	L	Highly Developed	Wheeled OHV 50" or <	43.56

Section 4

Explanation of Models

The aim of this project is to understand and predict the pattern of trail use recorded at multiple multi-use trails in the Bridger Mountains. We want to be able to predict trail use over a full calendar year and in locations where no trail counter data exist, within the Bridgers and nearby trails. Here, we provide a description of the statistical approach employed (generalized additive mixture modeling) in addition to brief overviews of modelling approaches that `gamm`s build upon (e.g. linear regression, generalized linear regression, and generalized additive models). In Section 5, we further this exploration in model choice by applying each to a single trail, Middle Cottonwood.

4.1 Linear Regression Model

Linear regression is used to model the linear relationship between a scalar or vector response and one or more explanatory variables.

$$y_i = \beta_0 + \beta_1 x_{i,1} + \beta_2 x_{i,2} + \ldots + \beta_{p-1} x_{i,p-1} + \epsilon_i.$$

where y_i is (are) the response variable(s) for each unit (i), $x_{i,p-1}$ are the explanatory variables, and β_{p-1} are the parameter coefficients. The errors, ϵ_i, are assumed to be normally distributed with mean 0 and constant variance σ^2. In this approach one would find estimates for the β_{p-1} parameters using values that minimize the sum of squared errors for the sample.

4.2 Generalized Linear Regression Model

In generalized linear models, the response variable y_i is now assumed to follow an exponential family distribution with mean μ_i, which is assumed to be some (often nonlinear) function of $x_i^T \beta$. Note that the covariates affect the distribution of y_i only through the linear combination $x_i^T \beta$.

The general form, written now in matrix multiplication format, is:

$$g(\mu) = \eta = X\beta$$
$$E(y) = \mu = g^{-1}(\eta)$$

where $g(.)$ is a link function relating the mean μ to the linear predictor(s) $X\beta$ (also denoted by η). Recall in linear regression we assume a Gaussian (i.e. normal) distribution for the response, we assume equal variance for all observations, and that there is a direct link of the linear predictor and the expected value μ, i.e. $\mu = X\beta$.

As such, the typical linear regression model is a generalized linear model with a Gaussian distribution and 'identity' link function.

For count data, a Poisson distribution is used. There is only one parameter to be considered, λ, since for the Poisson the mean and variance are equal. For the Poisson, the (canonical) link function $g(.)$, is the natural log, and so relates the log of λ to the linear predictor. As such we could also write it in terms of exponentiating the right-hand side.

$$\begin{gathered} y \sim \mathcal{P}(\lambda) \\ \ln(\lambda) = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 \ldots + b_p \cdot x_p \\ \lambda = e^{b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 \ldots + b_p \cdot x_p} \end{gathered}$$

Generalized linear models (and linear models as a subset) have a handful of tools to adapt to data that do not have quite so straightforward a relationship with associated covariates. Transformations to covariates that allow for inclusion of polynomial terms (e.g. quadratic, cubic, etc) are useful but have their own limits.

4.3 Generalized Additive Model

Generalized additive models (GAMs) are statistical models that can be used to estimate trends as smooth functions of time. This form allows for the now nonlinear predictor(s) to relate to the expected value, with whatever link function may be appropriate.

$$\begin{gathered} y \sim \text{ExpoFam}(\mu, \text{ etc. }) \\ E(y) = \mu \\ g(\mu) = b_0 + f\left(x_1\right) + f\left(x_2\right) \ldots + f\left(x_p\right) \end{gathered}$$

This approach is similar to GLM, but now instead of parametric coefficients on each of the variables, we now have smoothing functions (f) which are very flexible in their forms. Note that we can still include some covariates that do linearly relate to the response variable (y).

A spline is a function defined piece-wise by polynomials. Each consist of smaller basis functions, of which we may choose between several types/forms based on the data. We may also choose how many "pieces" to use by defining the number of knots (k) for each spline.

Spline Fun Fact: The term spline comes from the flexible devices used by shipbuilders and draftspersons to draw smooth shapes. Thanks, Wikipedia.

To model a potentially nonlinear smooth or surface, three different smooth functions are available:

- s() : for modeling a 1-dimensional smooth, or for modeling isotropic interactions (variables are measured in same units and on same scale)
- te(): for modeling 2- or n-dimensional interaction surfaces of variables that are not isotropic (but see info about d parameter below). Includes 'main' effects.
- ti(): for modeling 2- or n-dimensional interaction surfaces that do not include the 'main effects'.

4.3.1 Basis Functions

There are several smoothing bases b (splines) which are suitable for regression:

- thin plate regression splines
- cubic regression spline
- cyclic cubic regression spline
- P-splines

For a more in depth description of smooth terms as specified within a `gam` or `gamm` formula in R please refer to the associated Help document using the following code:

```
?mgcv::smooth.terms
```

In practice, R code for such a model may look like this:

```
gam_mod <- mgcv::gam(max.camera ~ max.count +
                       s(yday, bs = "cc") +
                       s(wday, bs = "cc",  k = 7) +
                       s(month, k = 3),
                     data = singleTrail,
                     knots = list(yday = c(0,365)),
                     family = poisson)
```

For each smoothing term you may select a basis function (the default is a thin plate regression spline (TPRS)) and an associated value for the number of knots, k.

4.3.2 Generalized Additive Mixture Model

Generalized additive mixed models (GAMMs) are an extension of generalized additive models widely used to model correlated and clustered responses. Temporal correlation in time series data may be accounted for by specifying various types of autoregressive correlation structures, via functionality already present in the separate nlme::lme() function, meant for fitting linear mixed models (LMMs). It is also possible to use lme4 in place of nlme as the underlying fitting engine, see gamm4 from package **gamm4**.

R code for GAMMs is very similar to that of GAMs, but now we add correlation structure (here, an AR(1) temporal correlation structure) to the model specification:

```
 ## AR(1)
gamm_AR1 <- mgcv::gamm(max.camera ~ max.count +
                         s(yday, bs = "cc") +
                         s(wday,
                           bs = "cc", k = 7) +
                         s(month, k = 3) +
                         data = singleTrail,
                       family = poisson,
                       knots = list(yday = c(0,365)),
                       correlation = corAR1(form = ~ yday)
)
```

4.3.3 Hierarchical Generalized Additive Models

Another natural extension to the GAM/GAMM framework is to allow smooth functional relationships between predictor and response to vary between groups, but in such a way that the different functions are in some sense pooled toward a common shape. With our application to hiking trails in the Bridger Mountains, we might be interested in understand how the relationship between trail use and various predictor variables differ between different trails (or subsections of trails).

Model structure for HGAMs varies depending on choices concerning global smoothers and how group-specific smoothers vary. Figure 4.1, originally published in Pedersen et al. (2019), shows the five types of models possible. In our application to all trail data in the Bridger Mountains (see Section 6) we focus on the three possible models that all include some form of a global smoother term, as models without this term are not well suited for prediction for trails not included as part of the training dataset.

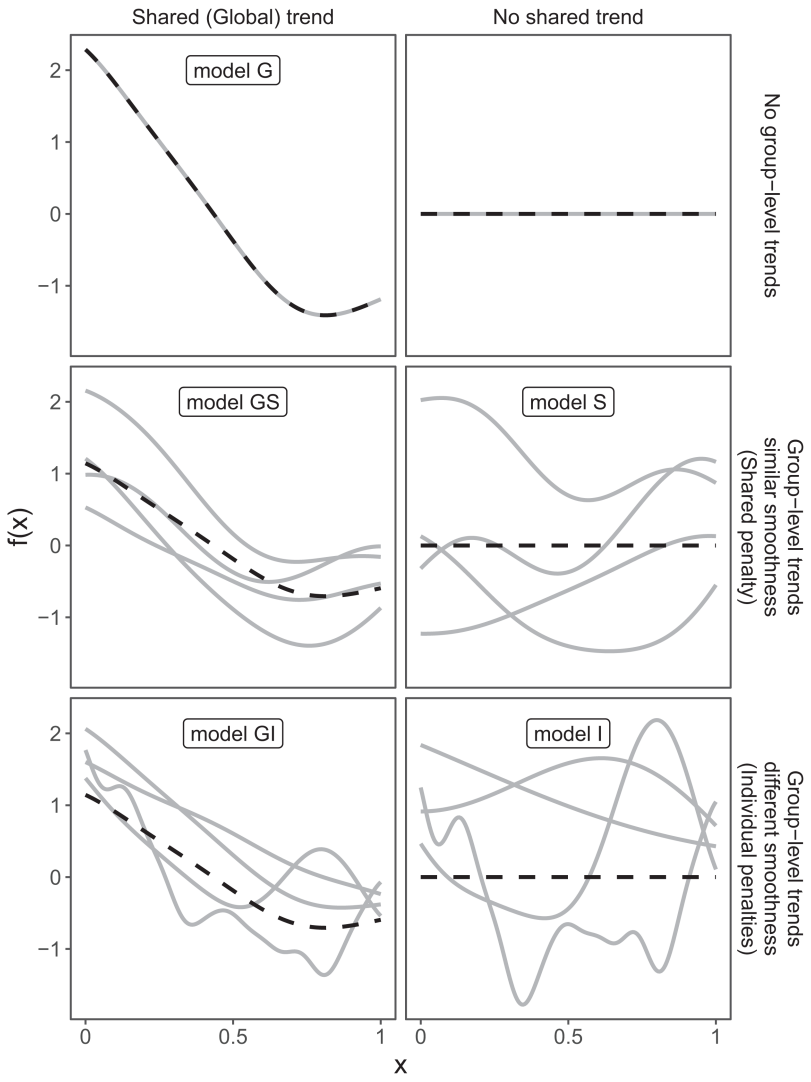


Figure 4.1: Alternate types of functional variation f(x) that can be fitted with HGAMs. Figure reproduced from: Pedersen EJ, Miller DL, Simpson GL, Ross N. 2019. Hierarchical generalized additive models in ecology: an introduction with mgcv. PeerJ 7:e6876 DOI: 10.7717/peerj.6876/fig-4

4.4 Furthur Explaination and Application of Models

All models covered in this summary overview are explored in more detail through an application to a single trail (Middle Cottonwood) in Section 5. Section 6 contains an analysis of all trails covered by trail-use cameras as well as a comparison of several HGAM model structures.

4.5 Additional Resources

This summary pulls heavily from the following resources that are very useful for a deeper dive into these models.

- Generalized Additive Models
- Simple overview of splines and basis functions

Section 5

Application to Middle Cottonwood Trail

In this section, we explore gradually more complicated model choices (each described in Section 4) which we apply to a single trail (rather than then entire network of trails) so we can focus on understanding the model applied to these data.

We begin with a quick look at the data available for our chosen trail, Middle Cottonwood Trail. Then we will progress through several `ga(m)m` models that incorporate more complexity with each iteration. This will include a detailed overview of useful diagnostic tools, a look at what explanatory predictor variables should/could be included, as well as an example of how we may forecast (predict into the near future) using our chosen model.

5.1 Data Used

This 2.3-mile out-and-back trail was monitored via infrared counters between 2021-06-27 and 2021-08-16. This daily time series count data is combined with data from Strava Metro, and local weather data.

Strava Metro partners with Headwaters Economics to provide aggregated (daily, monthly, or annually) trail use information for the Bridgers. These data are broken into multiple edges within a single trail. Eight such edges are included for Middle Cottonwood. These edges do not represent independent observations as they represent trail use on connected locations and thus likely reflect the same trail users continuing along the trail. We summarize these data to a single value using the edge ID with the maximum observed number of trail numbers for a given day.

In another partnership, AllTrails has provided information on daily trail name searches for a subsection of trails (including Middle Cottonwood) in the Bridger Mountains.

Thus each row in the dataset details the following:

- the number of trail users for a given day
- trail characteristics (trail name and number, location of trailhead)
- counter characteristics (counter ID number, owner, location along trail)
- Strava Metro information summarized to reflect the maximum count of all edges for a given day
- AllTrails information on search term numbers for a given day
- daily weather data (precipitation in inches, min/max temperature, AQI value, $PM_{2.5}$ concentration)

Figure 5.1 shows times series plot of counter and Strava trip counts, in addition to AllTrails search counts.

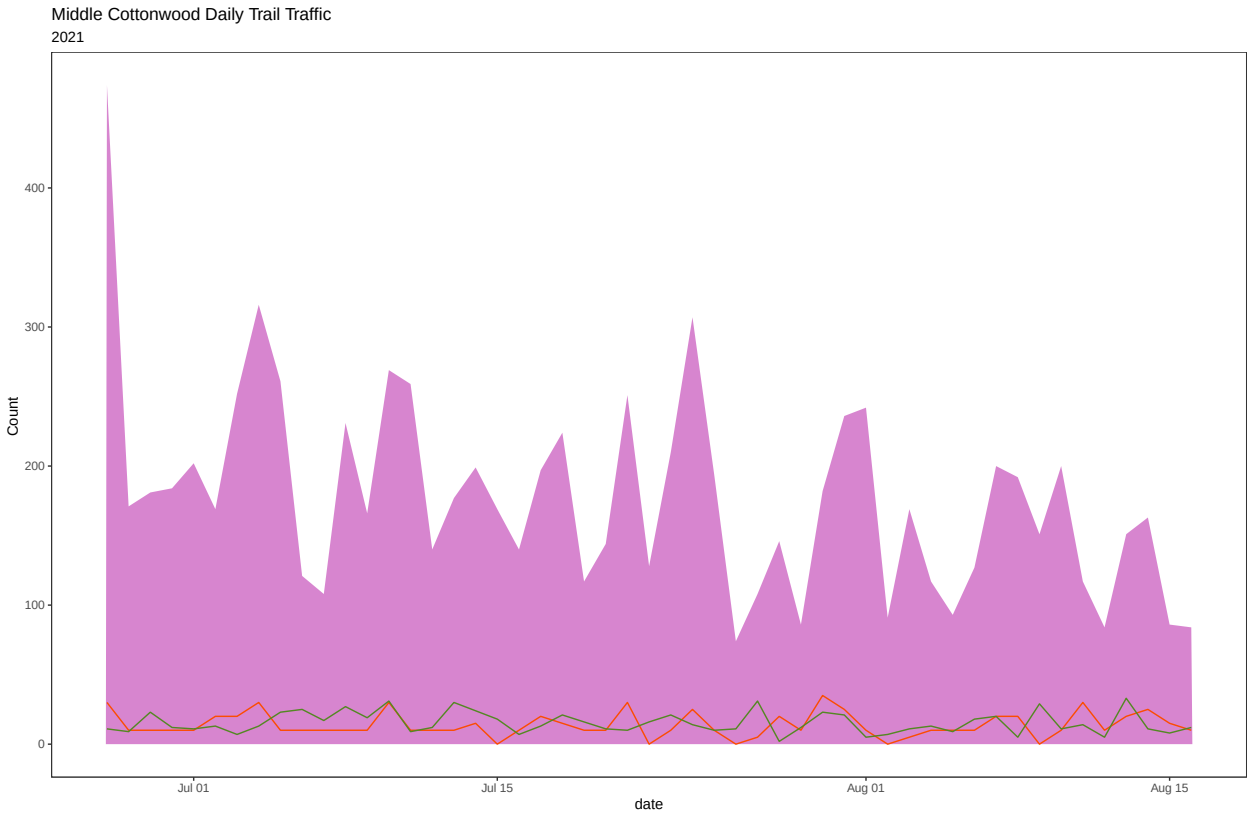


Figure 5.1: Daily trail traffic for Middle Cottonwood. Purple fill represents trail use counts from deployed counter cameras. Orange line represents trip counts per day provided by Strava Metro. Green line represents trail name searchs per day provided by AllTrails.

We also examine the relationship between our response variable (max.camera, or the number of trips counted on a trail section per day) and several explanatory variables (Figure 5.2).

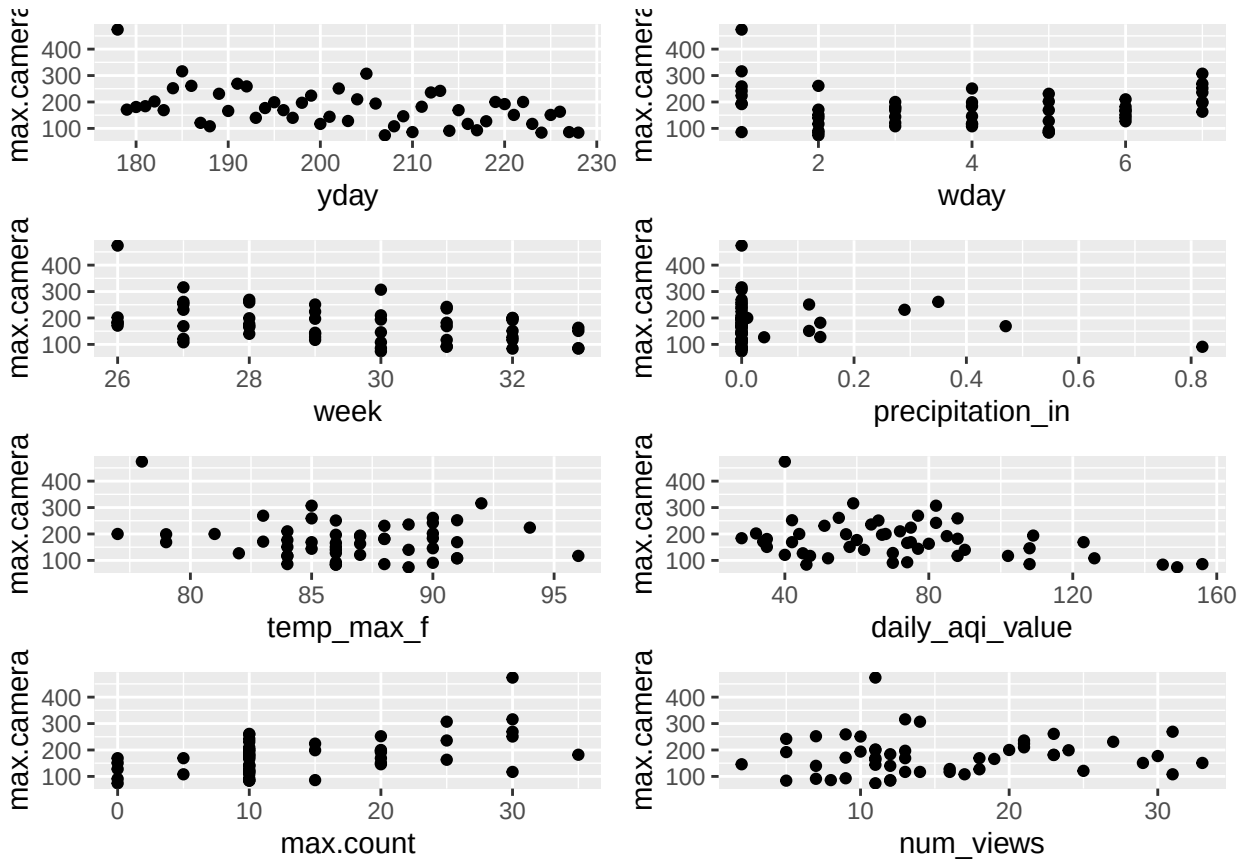


Figure 5.2: Relationship of select covariates to trail use for Middle Cottonwood.

5.2 Model Options

The following provides a very brief overview of several different types of regression models. This material is by no means an exhaustive coverage of the topic (see Section 4), but serves as an applied presentation allowing for us to build upon familiar modelling approaches towards what might be newer material.

All models are fit in R. GA(M)Ms are fit with the **mgcv** package (Wood, 2011, Wood et al. (2016), Wood (2004), Wood (2017), Wood (2003)).

5.2.1 Linear Regression Model

If we start simply and try to predict trail use count as our numeric response variable (y) via multiple linear regression (`lm`) using predictor variables, our model would have this general form:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + \epsilon$$

Translated in to `R` syntax, this model would look like:

```
lm_mod <- lm(max.camera ~
               max.count +
               num_views +
               daily_aqi_value +
               precipitation_in +
               temp_max_f,
             data = singleTrail)

save(lm_mod,
     file=here("output/models/singleT_lm.rda"),
     compress='xz')
```

This model assumes linear relationships between variables, a Gaussian (normal) distribution for the response variable, AND independent observations - **all of which are unlikely to be met with our data.**

5.2.2 Generalized Linear Regression Model

To address the issue of violated normality assumption in the previous model we can try a generalized linear model (`glm`) with a Poisson distribution (log link). The trail use count response is strictly non-negative and discrete (e.g. can't have 13.5 trail users). This model, using R syntax, would have this form:

```
glm_mod <- glm(max.camera ~
                 max.count +
                 num_views +
                 daily_aqi_value +
                 precipitation_in +
                 temp_max_f,
               data = singleTrail,
               family = poisson)

save(glm_mod,
     file=here("output/models/singleT_glm.rda"),
     compress='xz')
```

However, this does not address the non-linear relationship between response and predictors, nor the (temporal) dependence between observations.

5.2.3 Generalized Additive Models

We can address non-linearity through a simple generalized additive model (`gam`) with the general form,

$$y = b_0 + f_1(x_1) + f_2(x_2) \cdots + \epsilon$$

and the R syntax form of:

```
gam_mod <- mgcv::gam(max.camera ~
                       max.count +
                       num_views +
                       s(yday, bs = "cc") +
                       s(wday, bs = "cc",  k = 7) +
                       s(month, k = 3),
```

```
                      data = singleTrail,
                      knots = list(yday = c(0,365)),
                      method = 'REML',
                      family = poisson)

save(gam_mod,
     file=here("output/models/singleT_gam.rda"),
     compress='xz')
```

In the above code we fit a `gam` which is non-linear in `yday`, `wday`, and `month` but we also include `max.count` (Strava number of trips) and `num_views` (AllTrails searches) as linear terms. Note that these variable does not have the `s` prefix indicating a (spline based) smoothing term.

We specified basis functions for the day of year (`yday`) and day of week variable (`wday`). The code `bs="cc"` indicates that we are selecting a cyclic cubic regression spline, which is a penalized cubic regression spine whose ends match up. In the context of day of week we know that Sunday/Saturday would be the 'ends' that 'match' together. If we do not specify any, the default basis function is thin plate splines. They are the default smooth for `s` terms because there is a defined sense in which they are the optimal smoother of any given basis dimension/rank ((Wood, 2003)). One key advantage of this approach is that it avoids the knot placement problems of conventional regression spline modelling.

We have also specified the number of knots, k, for both wday and month. If we do not specify k then the default is 10. In both of these cases this results in the following errors:

> Error in place.knots(x, nk) : more knots than unique data values is not allowed
>
> Error in smooth.construct.tp.smooth.spec(object, dk$data, dk$knots) : A term has fewer unique covariate combinations than specified maximum degrees of freedom

(I believe the error messages are different due to differing basis functions.) For the `wday` variable we know that there's only 7 unique values (for each day of the week) and thus this is the highest value of k able to be used. Similarly, these data cover 3 `month` values and so that is our selection for k in this instance.

5.2.4 Generalized Additive Mixed Models

Recall that an assumption of the `gam` model is that observations are **conditionally** independent. Unless we specify that the data aren't independent using `gamm` instead of `gam`, the `gam` model will perform smoothness selection assuming that we have n independent observations. We first fit a `gamm` model to the exact same covariates and we used when fitting the previous `gam` model. The `R` syntax is as follows:

However, we may also include some more explanatory variables to better reflect our understanding of this system we are modelling. The following `gamm` model has additional non-linear terms included pulling from local weather information. We have specified `k=9` for `precipiation_in` again because the default value of 10 is too high and results in an error message and the code fails to run.

5.2.5 GAMM with additional temporal autocorrelation specification

If we model temporal autocorrelation through temporal terms in the model (as we did in `gamm_mod2`), then the smooth functions of temporal variables (such as `yday`, `wday`, etc) could be already accounting for the temporal structure in the data such that once we consider the model, the observations are independent. Note that it's also possible that the temporal autocorrelation is **not** captured through the terms in the model and additional complexity would be beneficial.

Autocorrelation is meant to capture either temporal or spatial dependence in a model. With a `ga(m)m` approach we consider both autocorrelation and a moving average (MA) process. An autoregressive process (AR(n)) is one in which the previous n points influence the current observations. A moving average (MA) process is one in which the current value is an average of preceding white noise. We are most interested in the main (fixed) effects of our model, while any autocorrelation is ancillary, but necessary to include.

For optimal choose of AR(p) and MA(q) orders `auto.arima` function, from the **forecast** package (Hyndman et al., 2022, Hyndman and Khandakar (2008)), is used on (normalized) residuals from the model fits. It automatically chooses optimal orders of ARMA (in our case) based on AIC criterion. As we can use just ARMA models in gamm, so nonstationarity isn't allowed, we set an argument stationary = TRUE.

```
##find values for p and q in the ARMA model
arma_res <- forecast::auto.arima(resid(gamm_mod2$lme,
                                       type = "normalized"),
                                 stationary = TRUE,
                                 seasonal = TRUE)

 #here there's no output bc an AR1 structure is sufficient, but in other instances
# this shows what values for p and q to choose
arma_res$coef
```

```
 ## AR(1)
gamm_mod2_AR1 <- gamm(max.camera ~
                        s(yday) +
                        s(month, k = 3) +
                        s(wday,
                          bs = "cc", k = 7) +
                        s(daily_aqi_value) +
                        s(temp_max_f, k = 5) +
                        s(precipitation_in, k = 5) +
                        max.count +
                        num_views,
              data = singleTrail,
              family = poisson,
              correlation = corAR1(form = ~ yday),
              method = "REML")
```

```
##
##  Maximum number of PQL iterations:  20
```

```
save(gamm_mod2_AR1,
     file=here("output/models/singleT_gamm-Ar1.rda"),
     compress='xz')
```

Even with the use of `auto.arima` we must still examine diagnostic plots (in this case, ACF and PACF plots for identifying autocorrelation) and summaries for each model (see Section 5.3 for more details).

5.2.6 Why not a Classic Time Series Analysis?

While both times series regression and GAM's can both be applied to temporal data, the two models are fundamentally different. Time series regression makes an assumption of a stationary series, while GAM deals with the trend internally with smoothing. With GAM, the estimated smoother is detrending the data and the (stationary) residuals are subjected to an ARMA model. The use of GAMs is useful when one is interested

in estimating the trends in time series data and those trends are, in general, non-linear. In classical time series modelling, the interest is in modelling data as stochastic trends using lagged versions of the response and/or current and lagged versions of a white noise process.

5.3 Diagnostic Practices

Once a model (or a suite of models) is fit, we need to take a more detailed look at model outputs to learn how to interpret the results of our model-fitting and better understand the relationships between variables.

5.3.1 Summarize

We start with the `summary()` function which may be applied to a `gam` object in R. Below we have the summary output for our `gamm` model with explanatory variables.

```
# summary(lm_mod)
# summary(glm_mod)
# summary(gam_mod)
# summary(gam_mod2)
# summary(gamm_mod$gam)
summary(gamm_mod2$gam)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## max.camera ~ s(yday, bs = "cc") + s(month, k = 3) + s(wday, bs = "cc",
##     k = 7) + s(daily_aqi_value) + s(temp_max_f) + s(precipitation_in,
##     k = 9) + max.count + num_views
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.943373   0.057037  86.670  < 2e-16 ***
## max.count   0.007160   0.002287   3.130  0.00581 **
## num_views   0.004407   0.003010   1.464  0.16051
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                        edf Ref.df      F  p-value
## s(yday)              5.955  8.000  5.103 0.000490 ***
## s(month)             1.000  1.000  8.798 0.008270 **
## s(wday)              4.621  5.000 30.473  < 2e-16 ***
## s(daily_aqi_value)   5.838  5.838  9.269 0.000119 ***
## s(temp_max_f)        6.485  6.485 13.780 1.14e-05 ***
## s(precipitation_in) 6.189  6.189 13.053 6.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.897
##   Scale est. = 1         n = 51
```

```
# summary(gamm_mod2_AR1$gam)
# summary(gamm_mod2_ARMA$gam)
```

The first part of the summary describes the model we fit. The "Family" component tells us the model assumes a Poisson distribution of our response, and the "Link" of "log" shows that the model transforms the predictions.

The next section describes the parametric terms of our model, referring to the linear terms in the model. This section may be familiar from linear modeling. It shows the coefficients for the linear terms in the model, their values, errors, test statistics, and p-values. Asterisks next to the p-values indicate statistical significance. In this case, the model intercept is significant, and the fixed effect of max.count (the number of aggregated/binned Strava trip counts) is also significant at the 0 level. However, the fixed effect of num_views (trail name searches in AllTrails) is not significant for Middle Cottonwood.

The next section covers smooth terms. For these smooths the summary for coefficients is not printed. This is because each smooth has coefficients for each basis function. Instead, the first column reads edf, which stands for effective degrees of freedom. This value represents the complexity of the smooth. An edf of 1 is equivalent to a straight line. An edf of 2 is equivalent to a quadratic curve, and so on, with higher edfs describing more wiggly curves. In our summary, all included smooth terms are significant at the 0.001 level. Note that the 'month' term has an edf of 1, indicating the smooth is equivalent to a straight line. The day of year term ('yday') has the highest edf (8.338) and thus highest wiggliness. These edf values (and wiggliness) for each smoothing term may be compared to the corresponding partial effects plot in Figure 5.3.

The terms to the right of the EDF column have to do with significance testing for smooths. The Ref.df and F columns are test statistics used in an ANOVA test to test overall significance of the smooth. The result of this test is the p-value to the right. It's important to note that these values are approximate, and it's important to visualize your model to check them.

The R-sq(adj.) does not a straight-forward interpretation of a measure of "proportion of variance explained" in nonlinear regression, and thus can not be employed as an absolute measure of model performance. Deviance explained should be a more generalized measurement of goodness of fit especially for non-gaussian models (such as we have here).

The scale estimate is $\hat{\phi}$, i.e. this is the value of ϕ estimated during model fitting. For the Poisson and Binomial families/distributions, by definition ϕ=1, but for other distributions this is not the case, including the Gaussian. In the Gaussian case, $\hat{\phi}$ is the residual standard error squared.

5.3.2 Visualize

After examining the summary, we should visualize our results. This may be accomplished using the several available function, however we choose to use the **gratia** package (Simpson, 2022) to produce these plots in **ggplot2** rather than base R.

5.3.2.1 Draw

```
# gratia::draw(gam_mod, residuals = T)
# gratia::draw(gam_mod2, residuals = T)
# gratia::draw(gamm_mod, residuals = T)
gratia::draw(gamm_mod2, residuals = T)
```

The plots (Figure 5.3 generated by **gratia**'s draw() (or **mgcv**'s plot() in base R graphics) function are partial effect plots. These may help diagnose problems with the model, such as oversmoothing. These plots show the component effect of each of the smooth or linear terms in the model, which add up to the overall prediction

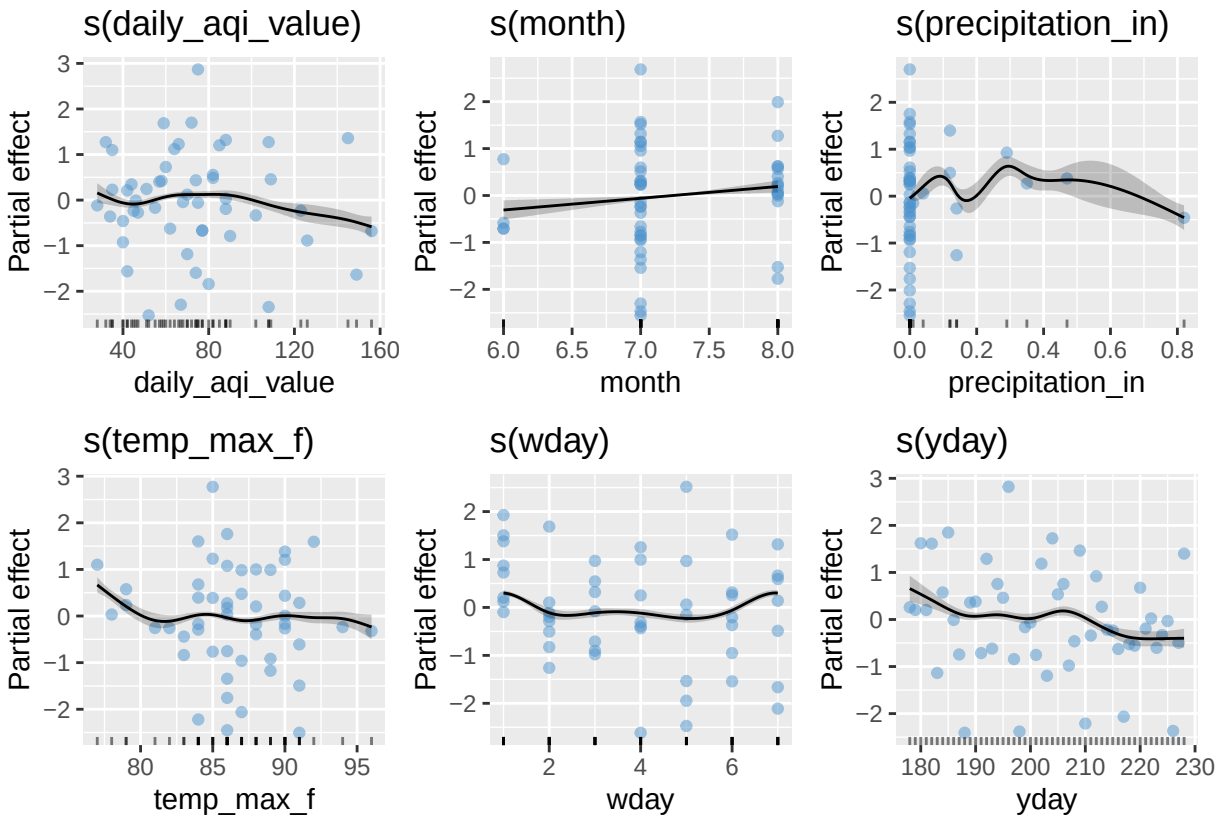


Figure 5.3: Plots of estimated smooths from the model fit with Middle Cottonwood trail use count data only.

(summed effect). We often want to show data alongside model predictions. These plots aid in this by (1) including covariate values along the bottom axis of the plots and (2) plotting partial residuals (here in light blue) on the plots. Partial residuals are the difference between the partial effect and the data, after all other partial effects have been accounted for. These plots also include shading representing 95% confidence interval for the mean shape of the effect.

5.3.2.2 GAM Check

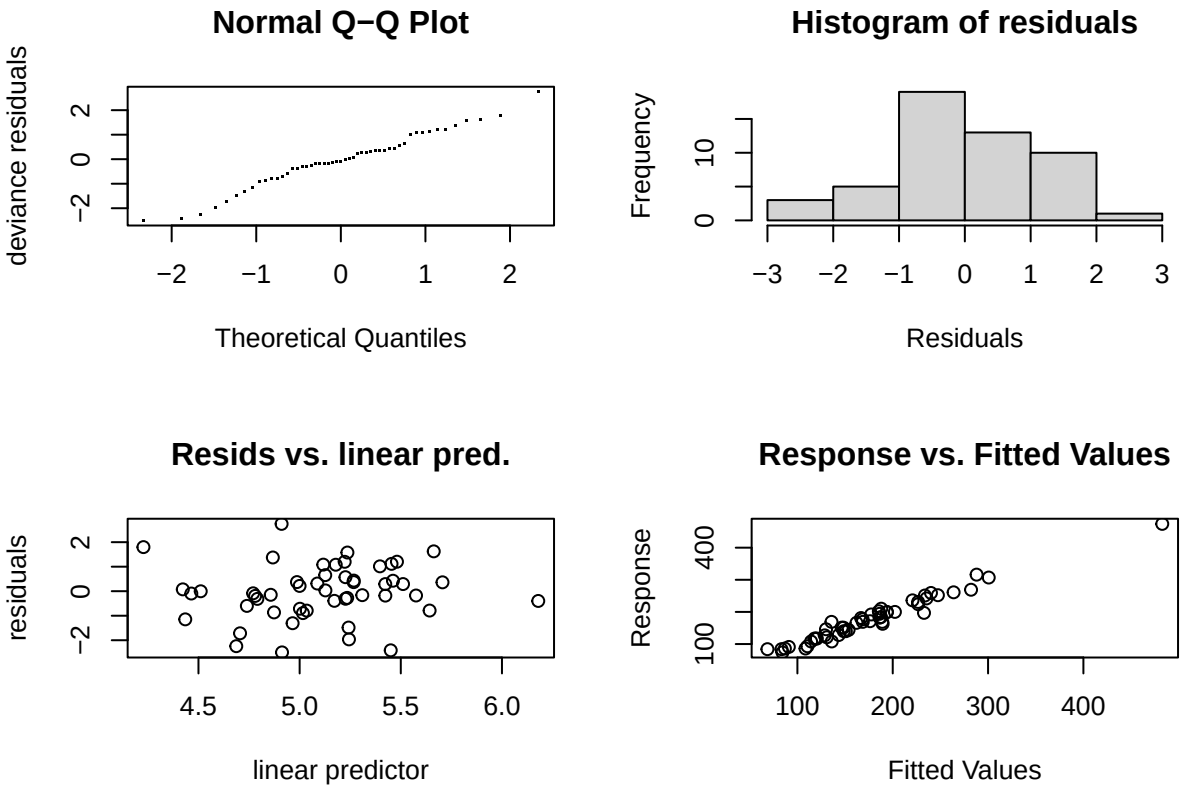
Running gam.check() on a model provides several outputs, in both the console and as plots. We'll start with the console output. First, gam.check() reports on model convergence. Here, it immediately provides a warning against interpreting these results when applied to a `gamm` based fit. We include this check here mainly to note this restriction and the usefulness if a `gam` fit is appropriate.

Below, we see a table of basis checking results. This shows a statistical test for patterns in model residuals, which should be random. Each line reports the test results for one smooth. It shows the k value or number of basis functions, the effective degrees of freedom, a test statistic, and p-value.

Here, small p-values indicate that residuals are not randomly distributed. This often means there are not enough basis functions.

This is an approximate test. Always visualize your results too, and compare the k and edf values in addition to looking at the p-value.

```
# gam.check(gam_mod)
# gam.check(gam_mod2)
layout(matrix(1:4, ncol = 2))
gam.check(gamm_mod2$gam)
```




```
##
```

```
## 'gamm' based fit - care required with interpretation.
## Checks based on working residuals may be misleading.
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                        k'  edf k-index p-value
## s(yday)              8.00 5.96    1.14    0.80
## s(month)             2.00 1.00    1.14    0.81
## s(wday)              5.00 4.62    0.87    0.17
## s(daily_aqi_value)  9.00 5.84    1.18    0.88
## s(temp_max_f)       9.00 6.49    1.16    0.85
## s(precipitation_in) 8.00 6.19    1.22    0.91
```

```
layout(1)
```

Each of these plot outputs gives a different way of looking at your model residuals. These plots show the results from the original, poorly fit model. On the top-left is a Q-Q plot, which compares the model residuals to a normal distribution. A well-fit model's residuals will be close to a straight line. On bottom left is a histogram of residuals. We would expect this to have a symmetrical bell shape. On top-right is a plot of residual values. These should be evenly distributed around zero. Finally, on the bottom-right is plot of response against fitted values. A perfect model would form a straight line. We don't expect a perfect model, but we do expect the pattern to cluster around the 1-to-1 line.

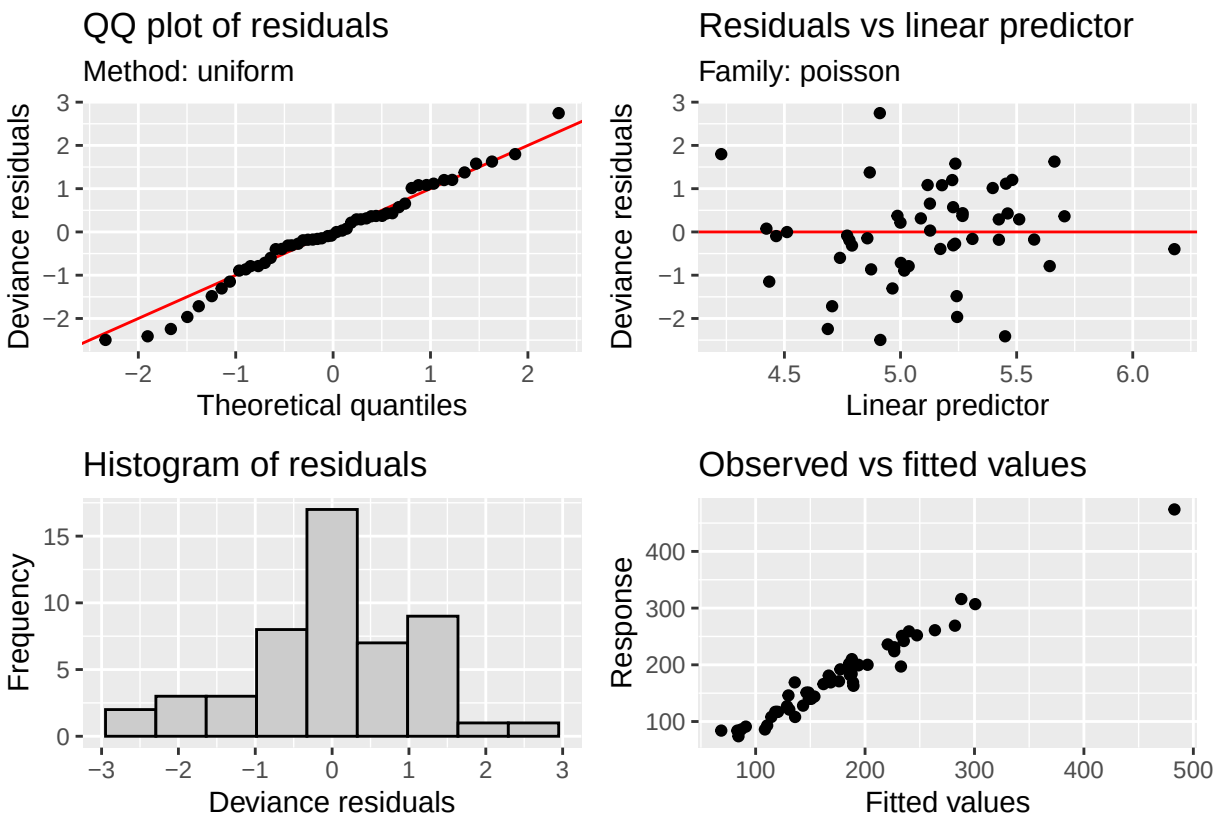
5.3.2.3 Appraise

The `appraise`() function provides another way to obtain standard diagnotic plots for GAMs.

The plots produced are (from left-to-right, top-to-bottom),

- quantile-quantile (QQ) plot of deviance residuals,
- scatterplot of deviance residuals against the linear predictor,
- histogram of deviance residuals, and
- scatterplot of observed vs fitted values.

```
# gratia::appraise(gam_mod)
# gratia::appraise(gam_mod2)
# gratia::appraise(gamm_mod$gam)
gratia::appraise(gamm_mod2$gam)
```

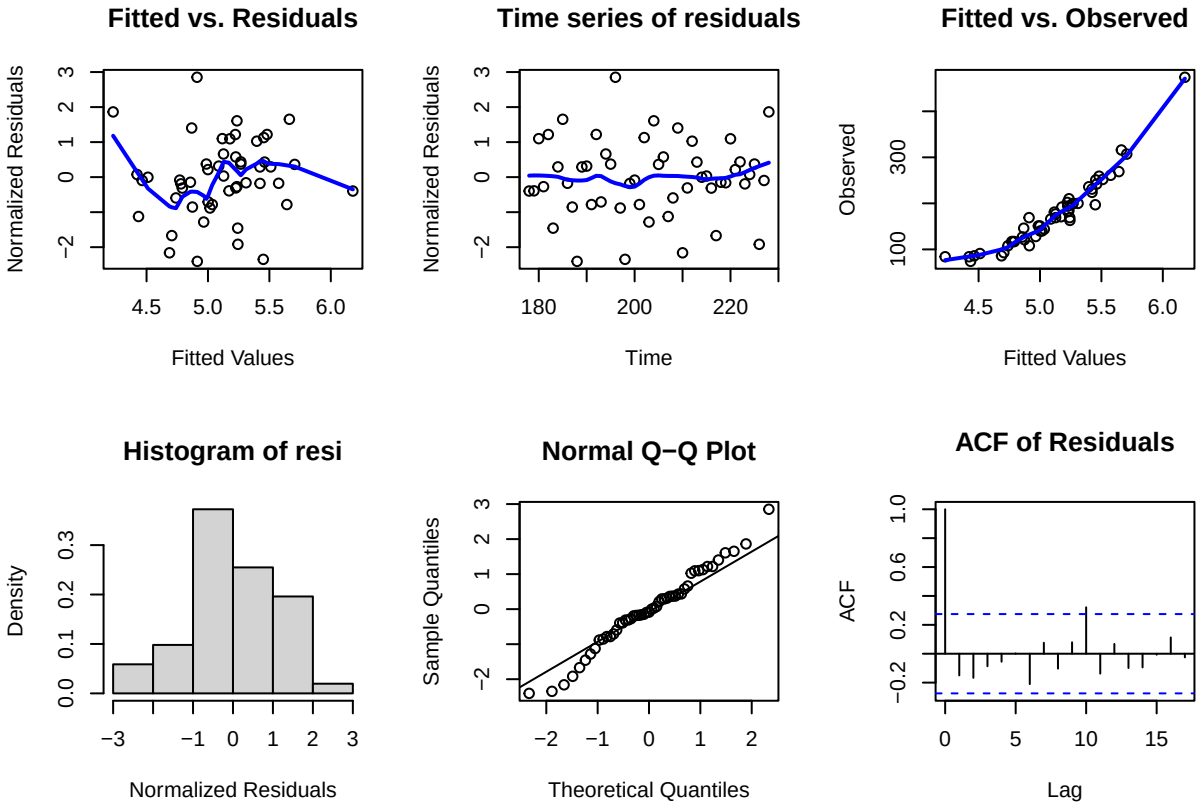


Fitted vs. response plots are generally less useful for non-normally distributed data as it can be difficult to visually assess if the observed data shows more heteroskedasticity than expected.

5.3.2.4 Time Series GAMM Diagnostic Plots

The following plots are a combination of various residuals plots and ACF plots produced with a bespoke R function created by Gavin Simpson and hosted on GitHub at: https://github.com/gavinsimpson/random_code/blob/master/tsDiagGamm.R. This function creates diagnostic plots specifically for time series data fit with a `gamm`, as we have with our trail use data.

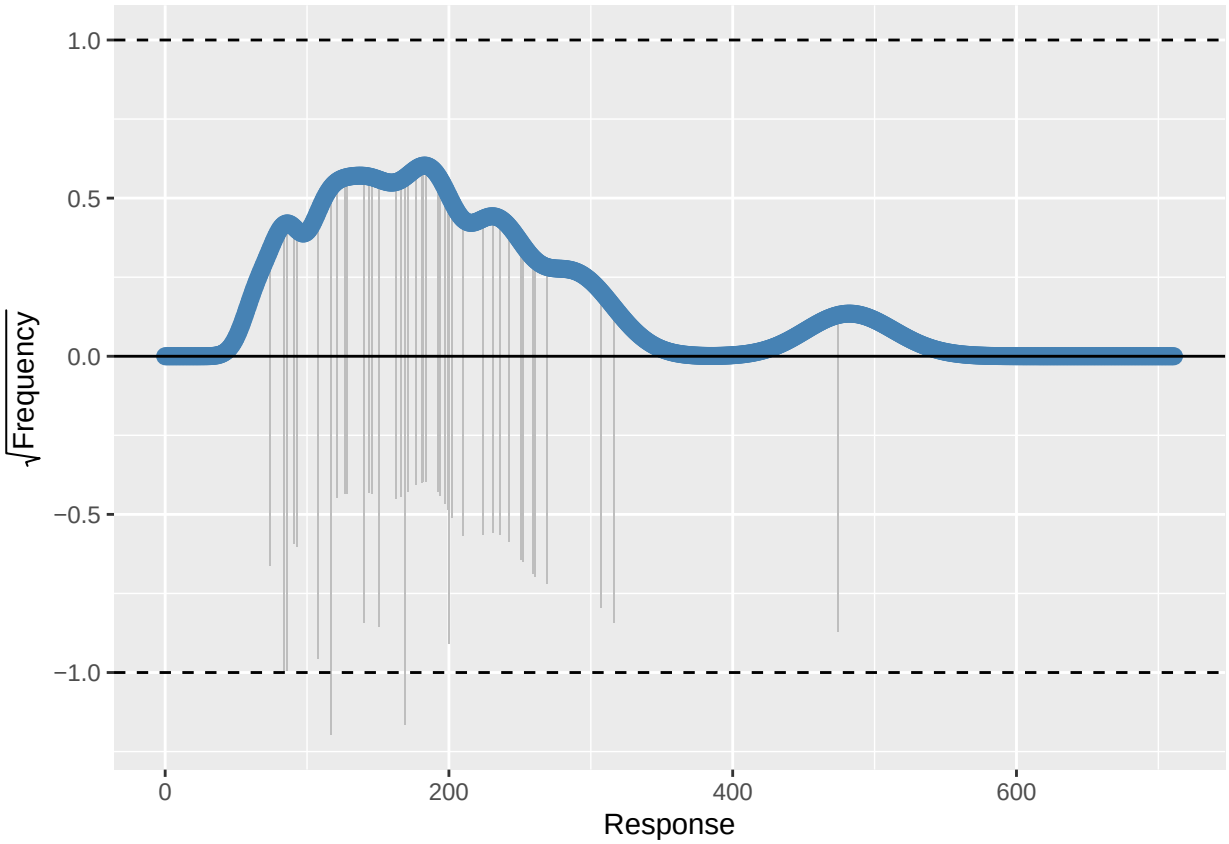
```
singleTrail_dropna <- singleTrail %>%
  tidyr::drop_na('daily_aqi_value')

with(singleTrail_dropna,
     tsDiagGamm(gamm_mod2,
                timevar = yday,
                observed = max.camera))
```



5.3.2.5 Rootogram

A good way to check how well the model compares with the observed data (and hence check for over-dispersion in the data relative to the conditional distribution implied by the model) is via a rootogram. A rootogram is a model diagnostic tool that assesses the goodness of fit of a statistical model. The observed values of the response are compared with those expected from the fitted model. For discrete, count responses, the frequency of each count (0, 1, 2, etc) in the observed data and expected from the conditional distribution of the response implied by the model are compared.

```
rg <- gratia::rootogram(gamm_mod2$gam)
draw(rg)
```



Looking at Figure 5.3.2.5 we see the main features of the rootogram:

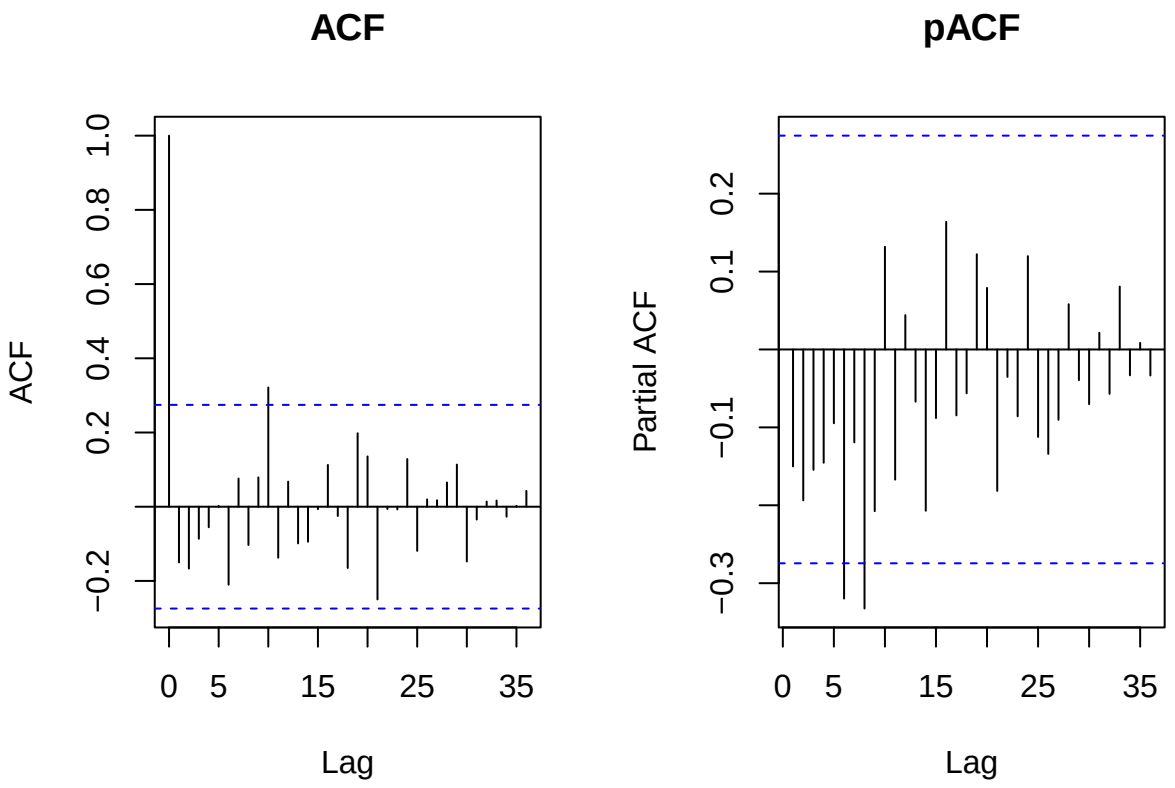
- expected counts, given the model, are shown by the thick blue line,
- observed counts are shown as bars, which in a hanging rootogram are show hanging from the blue line of expected counts,
- on the x-axis we have the count bin, 0 count, 1 count, 2 count, etc,
- on the y-axis we have the square root of the observed or expected count — the square root transformation allows for departures from expectations to be seen even at small frequencies,
- a reference line is drawn at a height of 0.

Because this is a hanging rootogram, we can think of the rootogram as relating to the fitted counts — if a bar doesn't reach the zero line then the model over predicts a particular count bin, and if the bar exceeds the zero line it under predicts.

5.3.2.6 Identifying AR and MA using ACF and PACF Plots

There are two visualizations of the residuals that can help you model autocorrelations: the ACF graph and the PACF. In general, ACF lets you assess the moving average component of the model and PACF lets you identify the autoregressive component. The p,q parameters can be estimated from the sharp cut off in the (P)ACF graphs.

```
layout(matrix(1:2, ncol = 2))
acf(resid(gamm_mod2$lme, type = "normalized"),
    lag.max = 36, main = "ACF")
pacf(resid(gamm_mod2$lme, type = "normalized"),
     lag.max = 36, main = "pACF")
```




```
layout(1)
```

5.3.3 Choosing an Appropriate Model for Trail Use in Middle Cottonwood

Often when we fit a linear regression model, we use R-squared as a way to assess how well a model fits the data.

R-squared represents the proportion of the variance in the response variable that can be explained by the predictor variables in a regression model. This number ranges from 0 to 1, with higher values indicating a better model fit. However, there is no such R-squared value for general linear models like logistic regression models and Poisson regression models. Instead, we can calculate a metric known as McFadden's R-Squared, which ranges from 0 to just under 1, with higher values indicating a better model fit.

You shouldn't compare the AICs between objects fitted with different software. gam() is fitted via the mgcv package, whereas gamm() fit is actually accomplished via the MASS (glmmPQL()) and then nlme (lme()) packages. It would be common for different constants to end up in the log likelihood.

Within the `gamm` framework we may use `anova`() to compare between models with different temporal autocorrelation structures.

```
##                    Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## gamm_mod2$lme          1 13 76.82418 101.9379 -25.41209
## gamm_mod2_AR1$lme      2 15 91.00384 119.9812 -30.50192 1 vs 2 10.17966  0.0062
```

This output indicates that our `gamm_mod2` fit (without any temporal correlation structure) is the best (i.e. has the lowest AIC and BIC values).

model	AIC	R.sq
lm_mod	571.6371	0.31
glm_mod	1192.8703	0.41
gam_mod	784.5881	0.64
gamm_mod	NA	0.64
gamm_mod2	NA	0.90
gamm_AR1	NA	0.83

Final Choice: Generalized Additive Mixture Model with Explanatory Variables

5.4 Prediction and Forecasting Trail Use

A common issue with any model, the generalized additive modelling framework included, is how to extrapolate beyond the range of data used to train the model. Temporal extrapolation is particularly tricky with temporal (time series) data. For GAMs, the issue arises as this framework uses splines to learn from the data via the basis functions. The splines are often set up directly related to the data included in the training set and it's not always clear how these should extend past that range of data, especially in this single trail application. (In Section 6 we apply this framework to a suite of trails and allow for a global smoother which can help inform trails with shorter camera deployment times by pulling/sharing from these global trends.)

Figure 5.4 shows predictions (and 95% credible interval on predicted values) for three different model specifications (`gam` with no temporal autocorrelation structure, `gamm` with no temporal autocorrelation structure, and `gamm` with an AR1 temporal autocorrelation structure). It's clear that all models perform best when interpolating, as expected. Improving forecasting with this single trail approach could be accomplished with several years of year-round observations.

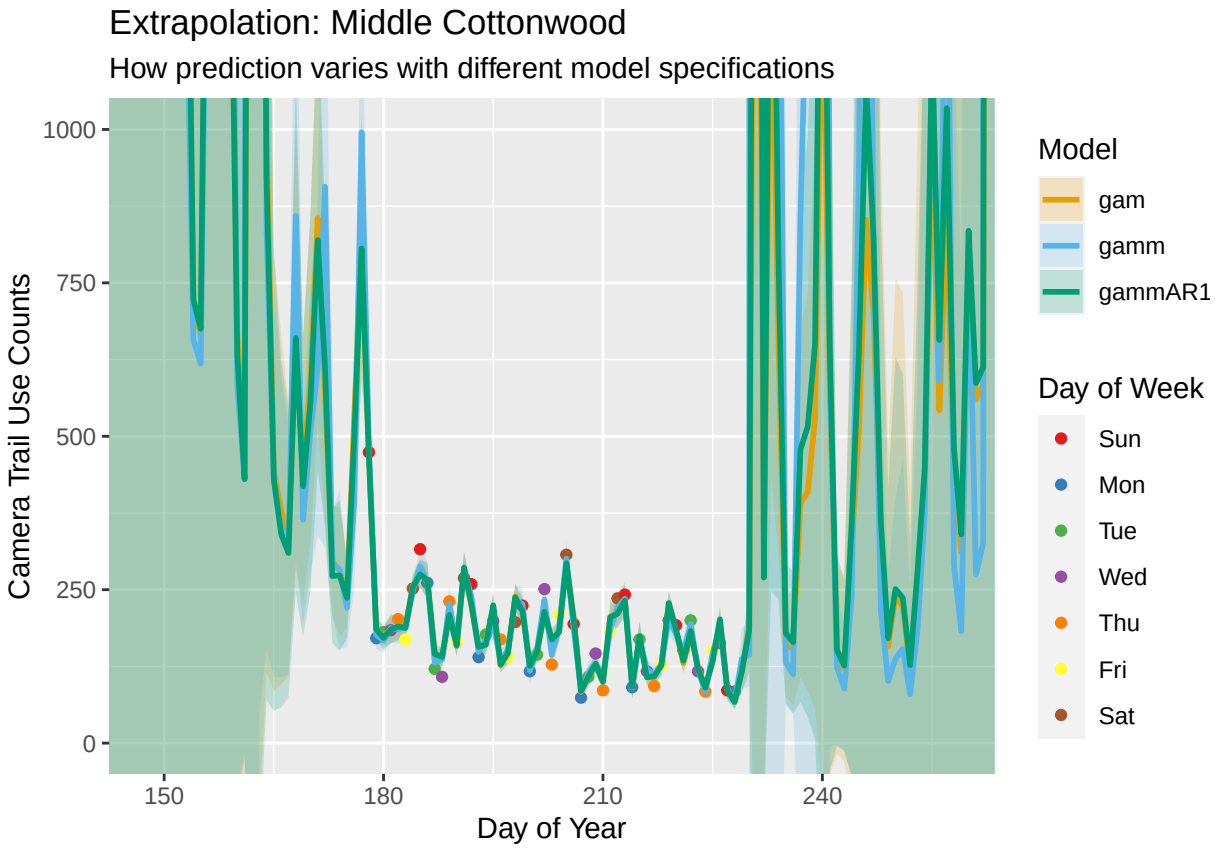


Figure 5.4: Predictions for Middle Cottonwood trail use showing interpolation and extrapolation behavior for temporal data range and extrended one month before and after data availability. Data points are colored by day of the week. Prediction lines (and error ribbons) are colored by model type applied.

Section 6

Analysis of All Trails

6.1 Data Used

This joint analysis covers the following Bridger Mountain subtrails:

1. Fairy Creek
2. College M
3. Bridger
4. Steep Way
5. Sacagawea Pass
6. Carroll Creek
7. Raptor View
8. Sypes Canyon
9. College M to Sypes
10. Truman Gulch
11. East Bridger South
12. East Bridger North
13. Lower Shafthouse
14. Corbly Gulch
15. North Cottonwood to Johnson Canyon
16. North Cottonwood Access
17. Ross Pass
18. Middle Cottonwood
19. Johnson Canyon Jeep Trail
20. Benchmark Road
21. Horsethief Mountain

6.2 Fitting a Generalized Additive Mixture Model

Here, we extend the Generalized Additive (Mixture) Model we first explained in Section 4 and then examined through application to Middle Cottonwood trail in Section 5 to now include all Bridger trails with counter camera data. This extension now includes different grouping levels (trail subsections) that require modeling of nonlinear functional relationships between covariates and outcomes where the shape of the function itself varies between different grouping levels. Hierarchical GA(M)Ms provide a natural extension to the standard GAM framework that allows smooth functional relationships between predictor and response to vary between groups, but in such a way that the different functions are in some sense pooled toward a common shape.

All models are fit in R. HGAMs are fit with the **mgcv** package (Wood, 2011, Wood et al. (2016), Wood (2004), Wood (2017), Wood (2003)).

In our application, where we want to estimate and predict trail use at trails in the Bridger Mountains, we may assume that each trail will have its own response function (e.g. melting snowpack may take longer at higher elevation and thus trail use would be concentrated on trails that clear sooner), but since the trails are all in the same range we can also expect similar responses over the year. Estimating a separate function for each trail (or subsection) throws away a lot of shared information and could result in highly noisy function estimates if there were only a few data points for some trails. Conversely, estimating a single average relationship could result in a function that does not predict any specific group well. We aimed for a hierarchical model that includes a global curve plus trail subsection-specific curves that were penalized to be close to the mean function.

Sometimes this dependence can be capture using the fixed and random effects in a GAMM, however when modelling all trails in this dataset we observed remaining temporal dependence (not shown, but see the ACF and PACF graph explanations in Section 5.3.2.6 for an overview of how to diagnose this dependence). For each model we examine three possible temporal autocorrelation structures: none, AR1, and ARMA. Summary and diagnostic plots are only presented for the best model.

6.2.1 Accounting for Trail/Subsection Grouping

A key feature of these data that we need to account for in our model is the random effect of trail, and for some trails the different subsections. Both `gam` and `gamm` in the **mgcv** package can incorporate random effects through a smoothing term that, in R, would look similar to

```
re_mod <- gam(response ~ maineffect +
                s(randomeffect, bs = 're'),
              data = data,
              method = 'REML')
```

Note that the random effect is denoted with `bs="re"`. Multiple random effects would each have its own smoothing term in the model. Another key requirement of this application is that the random effect term must be coded as a factor term. These random effect terms are adding random intercepts (deviations from the overall mean, the model constant term) for each level of the factors.

6.2.2 Global Smoothing Model (G)

We start with a simple GAMM structure with a single smooth for each of the variables.

In R we can write our model as:

```
gamm_modG_ARMA <- gamm(max.camera ~
                         s(yday, bs="tp") +
                         s(subsectionF, bs="re", k=21) +
                         s(month, bs="tp" k = 7) +
                         s(wday, bs = "cc", k = 7) +
                         s(daily_aqi_value, bs="tp", k = 10) +
                         s(temp_max_f, bs="tp", k = 10) +
                         s(precipitation_in, bs = "tp", k = 5) +
                        s(totallength_miles,  bs="tp", k = 10) +
                        total_traveltime +
                         max.count
                        ,
```

```
                        method = 'REML',
                         data = allTrail,
                        correlation = corARMA(form = ~yday|subsectionF,
                                              p = 1, q = 2),
                        family = poisson,
                        niterPQL = 20)
```

The arguments to the s() terms are smoothed. For each we explicitly specify the type of smoother to be used with the bs argument, and the maximum number of basis functions with k. The default type of smoother is the TPRS smoother ("tp") and the default value for k (for TPRS) is 10. We use a cyclic cubic spline ("cc") for day of week (wday) and set k=7 as we have seven unique values in this variable. We also set k=7 for month for the same reason; the data spans seven months of the year. If camera counters are deployed year-round in the future we would use bs="cc" and k=12 for this variable. The random effect smoother (bs="re") that we used for the subsectionF factor always has a k value equal to the number of levels in the grouping variable (here, 21). We restrict the number of knots for `precipitation_in` (k=5) to curb some weird behavior likely due to very few non-zero values.

For each model (G, GS, GI) we will use the **forecast** package to determine the optimum values of p and q in the ARMA correlation structure, but, for brevity, won't always report this outcome.

```
## this should help find values for p and q in the ARMA model
arma_res_G <- forecast::auto.arima(resid(gamm_modG$lme, type = "normalized"),
                                   seasonal = T)

arma_res_G$coef
```

```
##        ar1        ar2        ma1
##  1.1270770 -0.1848592 -0.7771053
```

The summary and various diagnostic plots will only be shown for the "best" model (between the different correlation structures) as determined by anova. For model G, we will present the model with the ARMA correlation structure only.

```
##                     Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## gamm_modG$lme           1 15 39089.53 39164.07 -19529.76
## gamm_modG_AR1$lme       2 16 33455.64 33535.16 -16711.82 1 vs 2 5635.882  <.0001
## gamm_modG_ARMA$lme      3 20 31217.28 31316.68 -15588.64 2 vs 3 2246.360  <.0001
```

The summary output indicates that all of our included parametric (linear) coefficients and smooth terms are (approximately) significant. The effective degree of freedom (edf) values represents the complexity of the smooth. We can see in this output that the highest smooth complexity is for `subsectionF`. The adjusted R-sq value (which should not be employed as an absolute measure of model performance) is 0.78 for this model.

```
summary(gamm_modG_ARMA$gam)
```

```
##
## Family: poisson
## Link function: log
##
## Formula:
## max.camera ~ s(yday, bs = "cc") + s(subsectionF, bs = "re", k = sub.number) +
```

```
##     s(month, k = 7) + s(wday, bs = "cc", k = 7) + s(daily_aqi_value) +
##     s(temp_max_f) + s(precipitation_in, k = 5) + s(totallength_miles) +
##     total_traveltime + max.count
##
## Parametric coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.3383574  0.3398840   9.822  < 2e-16 ***
## total_traveltime -0.0256714  0.0095960  -2.675  0.00759 **
## max.count         0.0103780  0.0005349  19.401  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                          edf Ref.df       F  p-value
## s(yday)                7.966  8.000 582.017  < 2e-16 ***
## s(subsectionF)        16.262 19.000  95.121  < 2e-16 ***
## s(month)               5.209  5.209  25.987  < 2e-16 ***
## s(wday)                4.984  5.000 627.438  < 2e-16 ***
## s(daily_aqi_value)     8.888  8.888 166.946  < 2e-16 ***
## s(temp_max_f)          8.747  8.747 878.221  < 2e-16 ***
## s(precipitation_in)    3.797  3.797  71.549  < 2e-16 ***
## s(totallength_miles)   2.491  2.491   8.491 0.000144 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.784
##   Scale est. = 1         n = 1064
```

In figures 6.1 and 6.2, we examine several diagnostic plots for a time series GAMM fit. A more detailed explanation of all of these plots and how to interpret them is available in Section 5.3. We are not looking for a perfect fit with this model, rather including these diagnostic plots so that they may be compared to those from fitting the data to models GS and GI.

Figure 6.3 partial effects plots for model G with an ARMA(1,2) temporal correlation structure. Partial effects are the isolated effects of one particular predictor or interaction of predictors. The output now includes a QQ-plot for the random effects term, showing the estimated intercepts for the different levels of `subsectionF`.

```
gratia::draw(gamm_modG_ARMA$gam, residuals = F)
```

The rootogram plot in Figure 6.4 shows that the data are slightly overdispersed (the variance, which is expected to be the same as the mean, is larger). Overdispersion for a poisson family distribution is not uncommon in real world data. Alternative families may be specified in the `gamm` code through the family argument, such as quasipoisson, negative binomial or (for `gam` only) a zero-inflated poisson family. Several alternatives were explored (not shown) but none provided improvements in overdisperson.

```
rg <- gratia::rootogram(gamm_modG_ARMA$gam)
draw(rg)
```

Averaging over all of the variation (between trails) results in a relatively imprecise (diffuse) estimates of trail use (Figures 6.5 and 6.6), and viewing species-specific plots of observed vs. predicted values (Figure 6.7), it is apparent that the model fits some of the trail sections better than others. This model could potentially be improved by adding intergroup variation between trail subsections.

Model GS is able to effectively capture the obseved pattern of trial use variation between trail subsections and shows slightly less evidence of overdispersion (Figure 6.11) in some trail subsections compared to Model G (note the difference in Corbly Gulch).

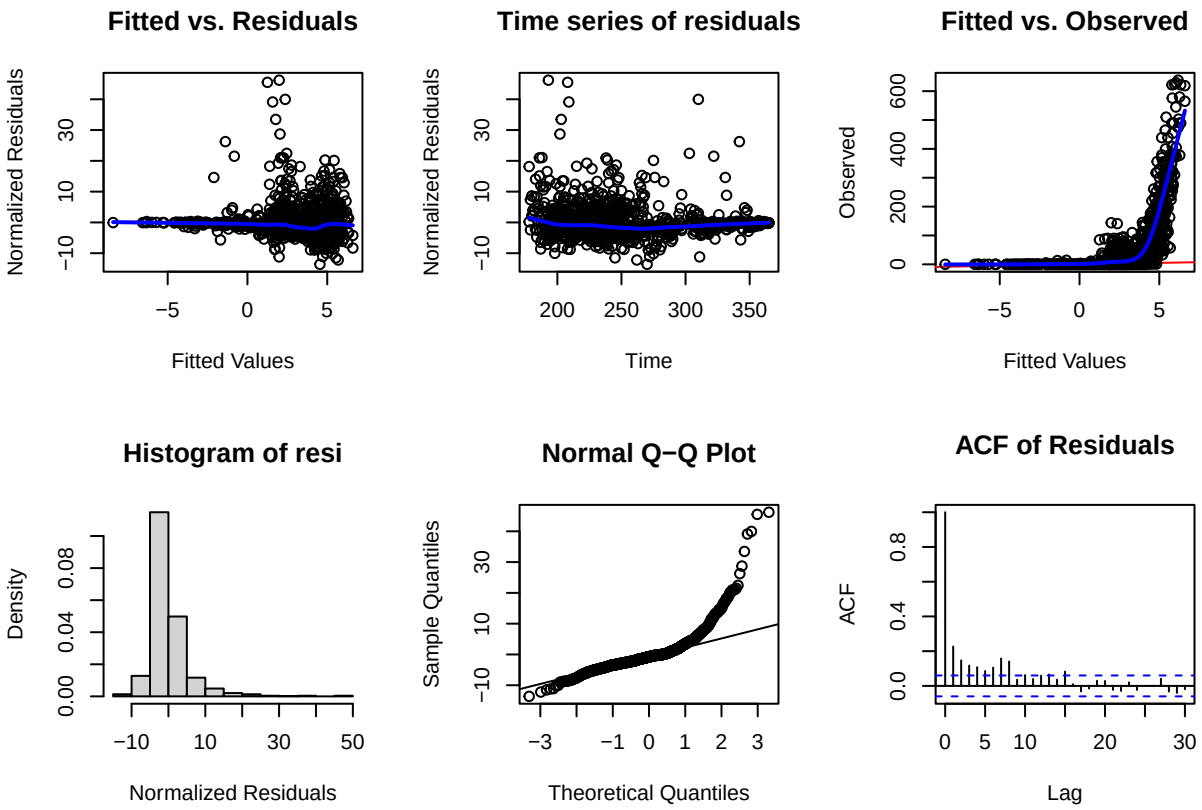


Figure 6.1: GAMM time series diagnostic plots for model G.

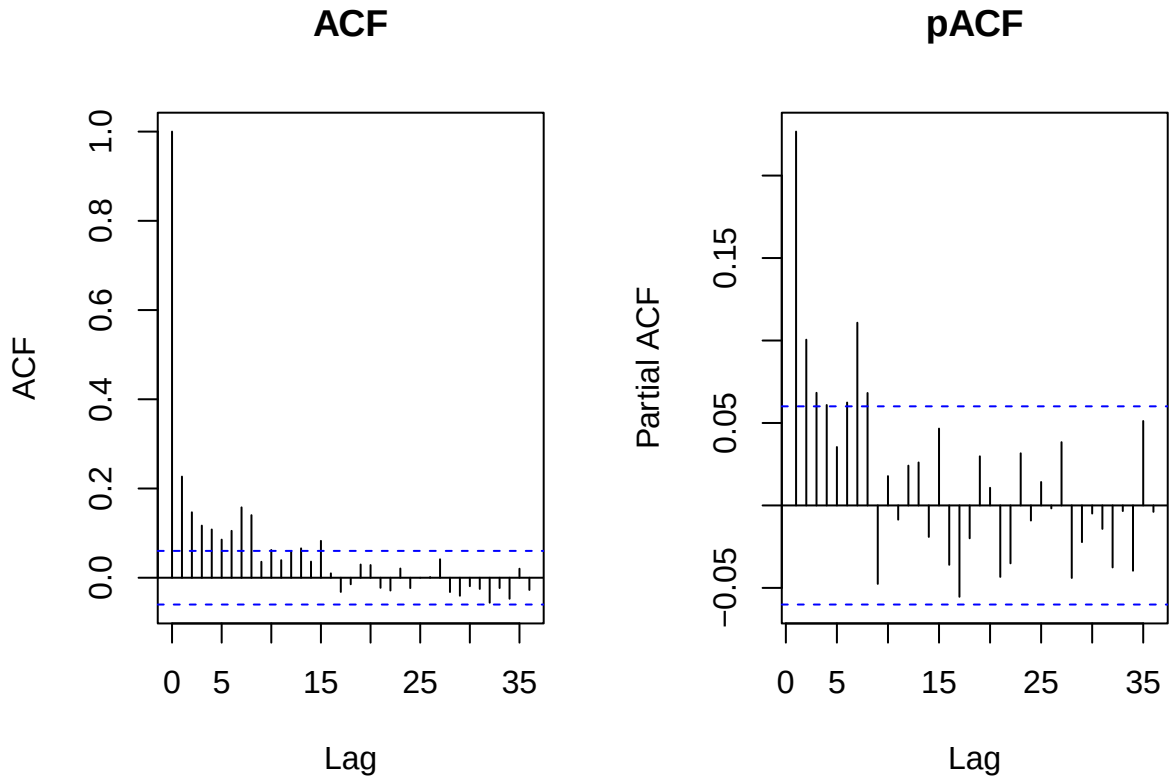


Figure 6.2: ACF/pACF diagnostic plots for model G.

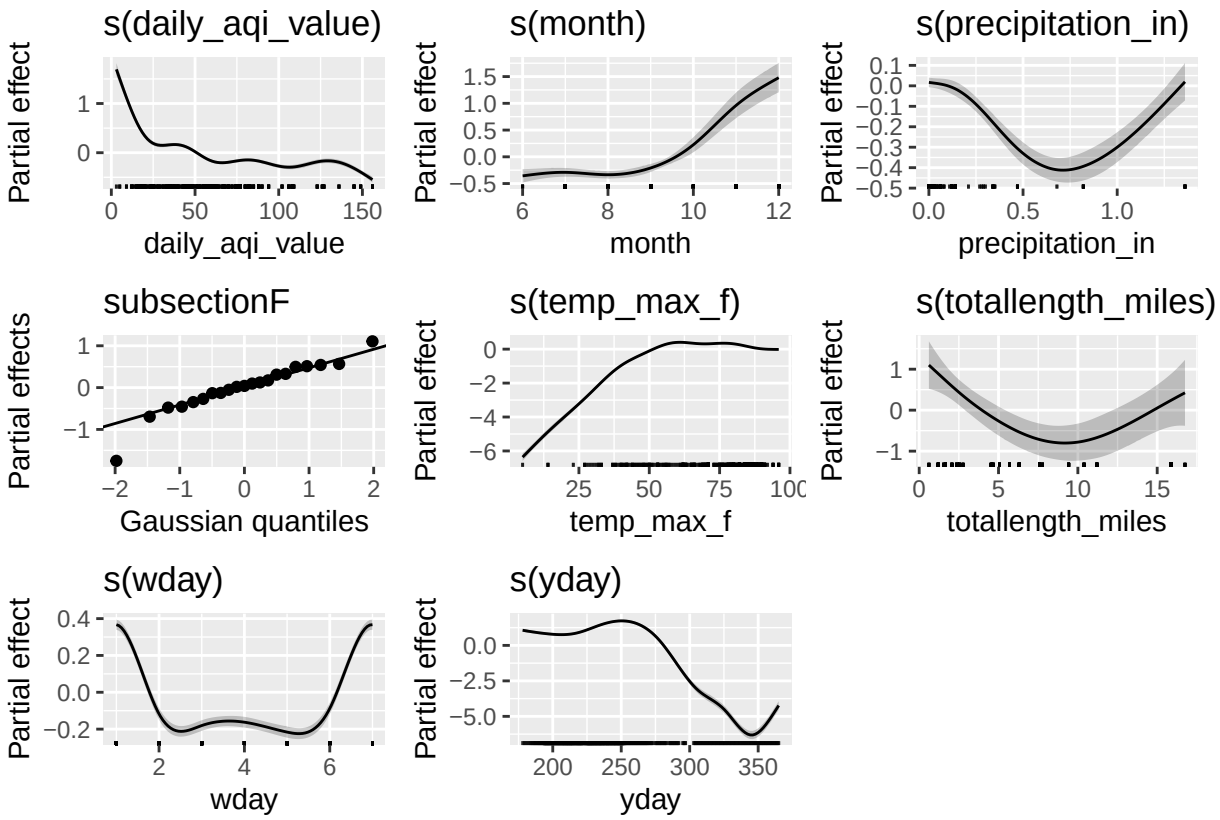


Figure 6.3: Partial effects plots for model G.

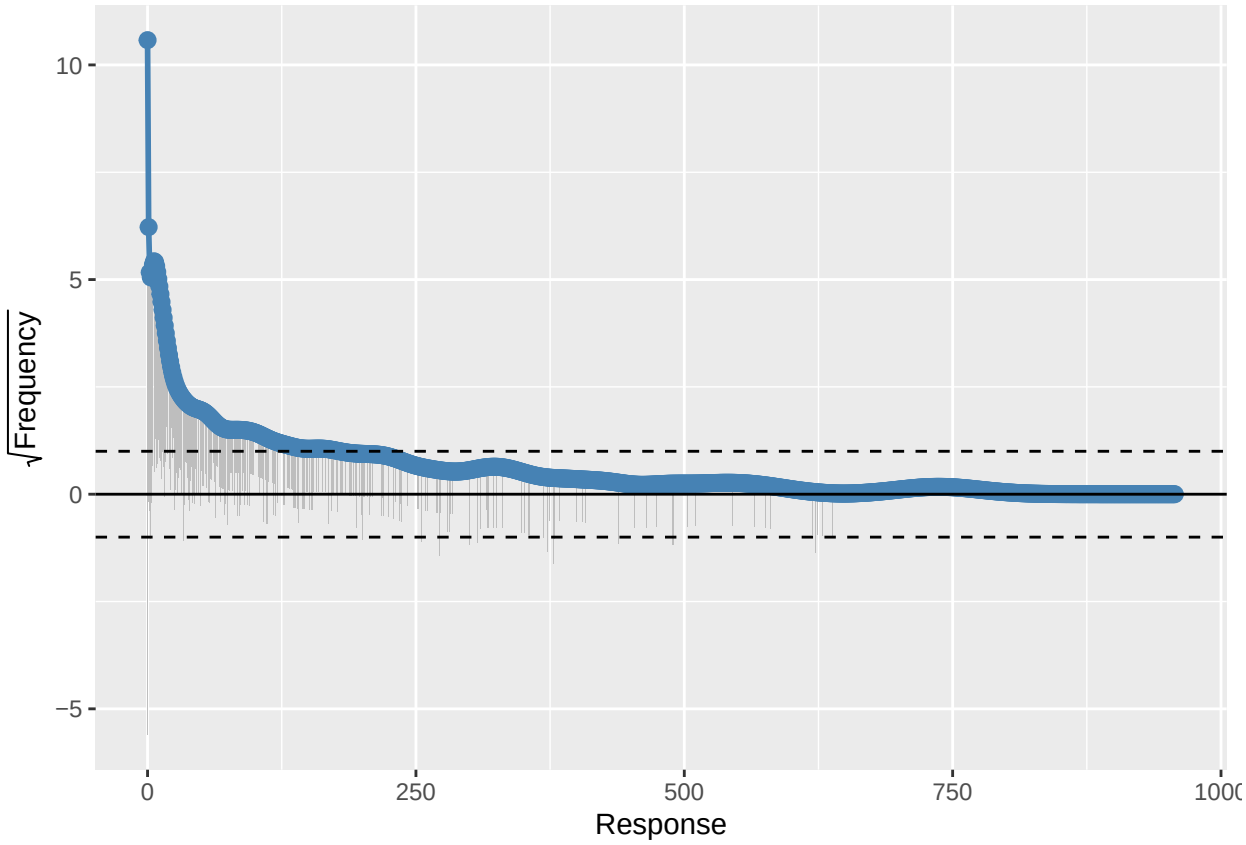


Figure 6.4: Rootogram for checking for overdispersion.

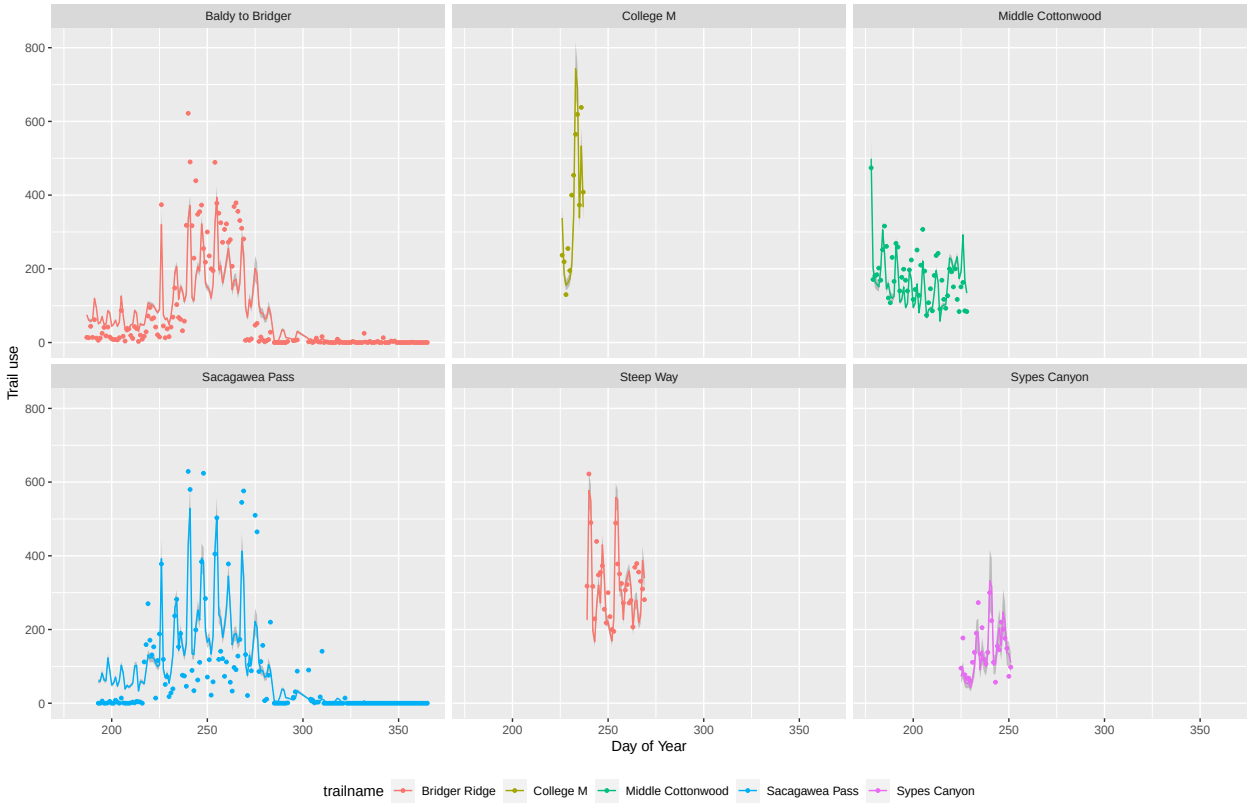


Figure 6.5: Predicted trail use count values (lines) versus observed trail use (points) for each high-use trail subsection, based on model G.

Figure 6.6: Predicted trail use count values (lines) versus observed trail use (points) for each low-use trail subsection, based on model G.

```
#add the predicted values from the model
allTrail_G_pred <- transform(allTrail_G,
                        mod_G = predict(gamm_modG_ARMA$gam,
                                        type = "response"))

ggplot(allTrail_G_pred, aes(x=mod_G, y=max.camera)) +
  facet_wrap(~subsectionF, ncol= 3) +
  geom_point(alpha=0.3, aes(color = trailname)) +
  scale_color_manual(name = "Trail", values = colors) +
  geom_abline() +
  theme(legend.position="none") +
  labs(x="Predicted count", y="Observed count")
```

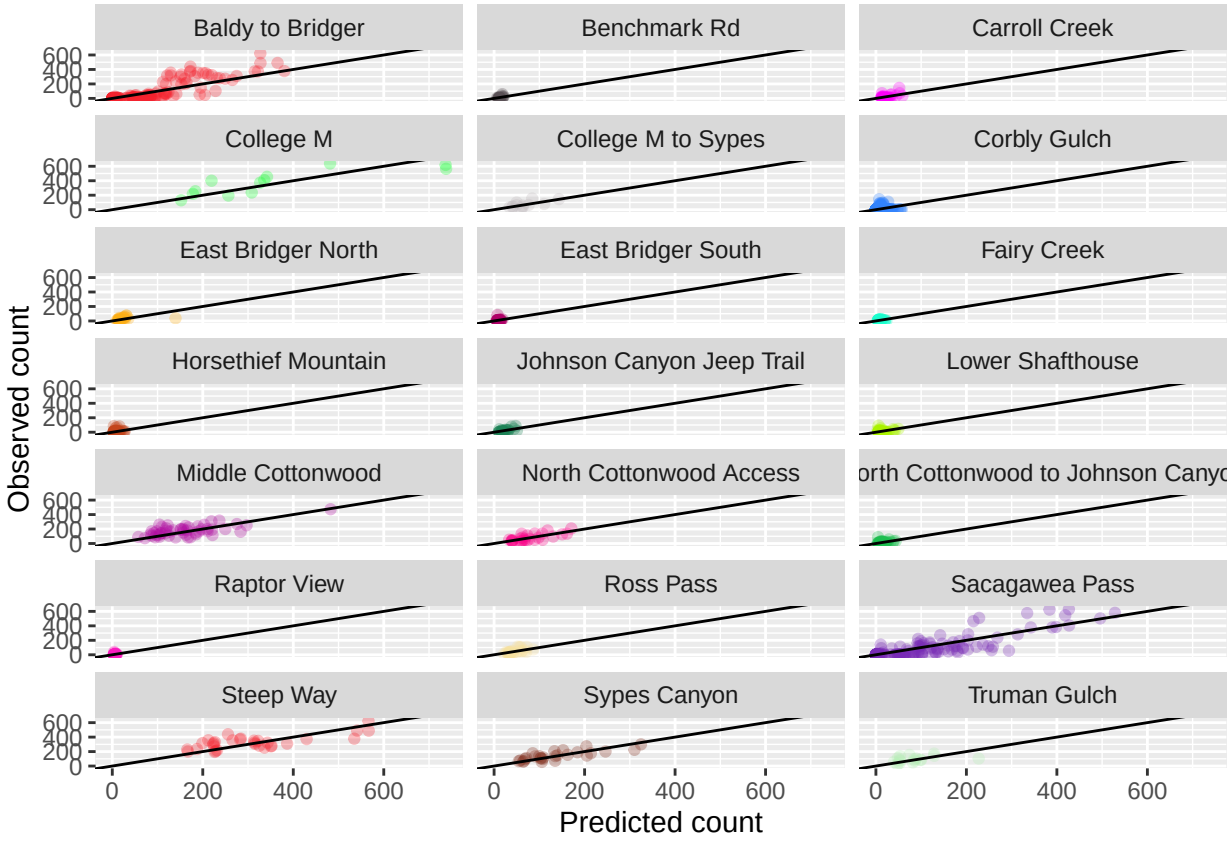


Figure 6.7: Assuming a well-fitted model G, we would expect all trail subsections exhibiting similar patterns of dispersion around the 1-1 line (and as we are assuming the data is Poisson, the variance around the mean should equal the mean). Instead we see that variance around the predicted value is much higher for some trails such as Sacagawea Pass.

6.2.3 Single common smoother plus group-level smoothers that have the same wiggliness (model GS)

Model GS constricts all groups to having similar functional responses, but, unlike model G, intergroup variation in responses is allowed. This approach works by allowing each grouping level (here, `subsectionF`) to have its own functional response, but penalizing functions that are too far from the average.

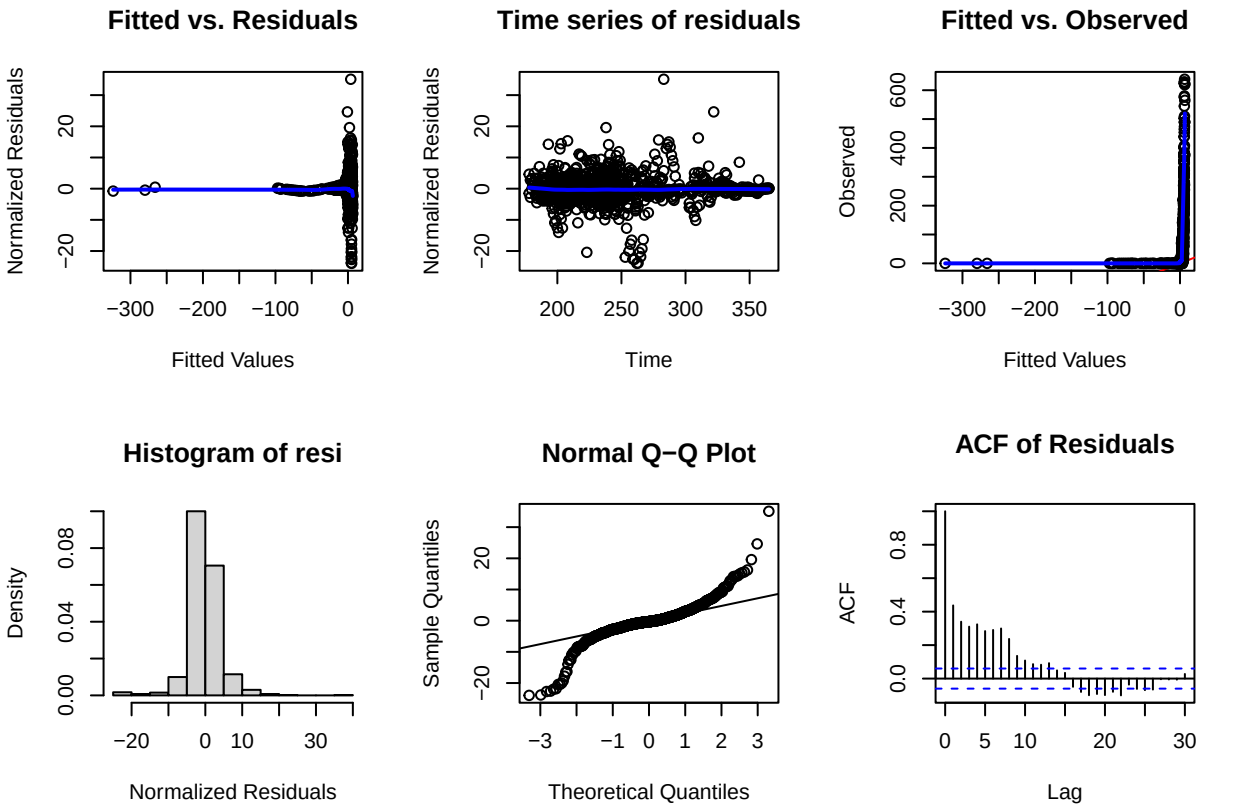
In R we can write our model as:

```
gamm_modGS_AR1 <- gamm(max.camera ~
                         s(yday, m=2, bs="cc") +
                         s(yday, subsectionF,
                           m=2, bs="fs", k = 21) +
                         # s(subsectionF, bs = "re", k= 21) +
                         s(month, bs = "cc", k = 7) +
                         s(wday,
                           bs = "cc", k = 7) +
                         s(daily_aqi_value) +
                         s(temp_max_f) +
                         s(precipitation_in, k = 5) +
                        s(totallength_miles) +
                       total_traveltime +
                           max.count,
                        knots = list(yday = c(0,365),
                                     month = c(0, 13)),
                       data = allTrail,
                       method = "REML",
                       correlation = corAR1(form = ~yday|subsectionF),
                       family = poisson)
```

With this model specification we explicitly specifying one term for the global smoother (as in model G, above) then added a second smooth term specifying the group-level smooth terms (here, `subsectionF`), using a penalty term that tends to draw these group-level smoothers toward zero. This penalty is incorporated via the factor-smoothing basis type (`bs = "fs"`) which creates a copy of each set of basis functions for each level of the grouping variable, but only estimates one smoothing parameter for all groups (see `?mgcv::factor.smooth.interaction` for details).

Model GS with an ARMA correlation structure has been selected, but the summary does not seem to work on this model fit object. However, all other diagnostic plots and predictions work.

Diagnostic plots for time series data show there still is temporal autocorrelation left in the model.



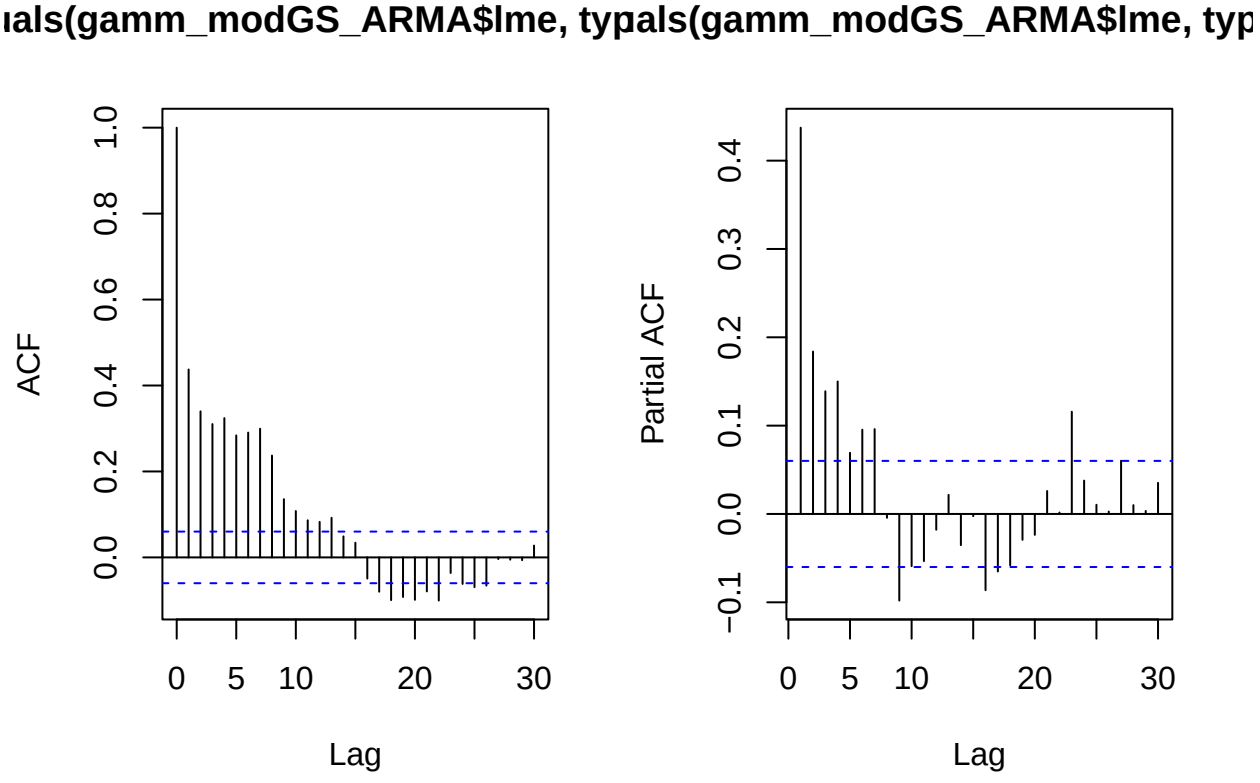


Figure 6.8 shows the fitted smoothers for `gamm_modGS_ARMA_sub`. The plots of group-specific smoothers (bottom left) indicate that trail subsections differ not only in average (log) trail use (which would correspond to each trail having a straight line at different levels for the group-level smoother), but differ slightly in the shape of their functional responses. Figures 6.9 and 6.10 shows how the global and group-specific smoothers

combine to predict trail use for individual trails. We see that, unlike in the single global smoother case above, none of the curves deviate from the data systematically. Some trails with notable improvement in prediction include Corbly Gulch and Middle Cottonwood. Sacagawea Pass still seems to have the highest deviation between observed and predicted values.

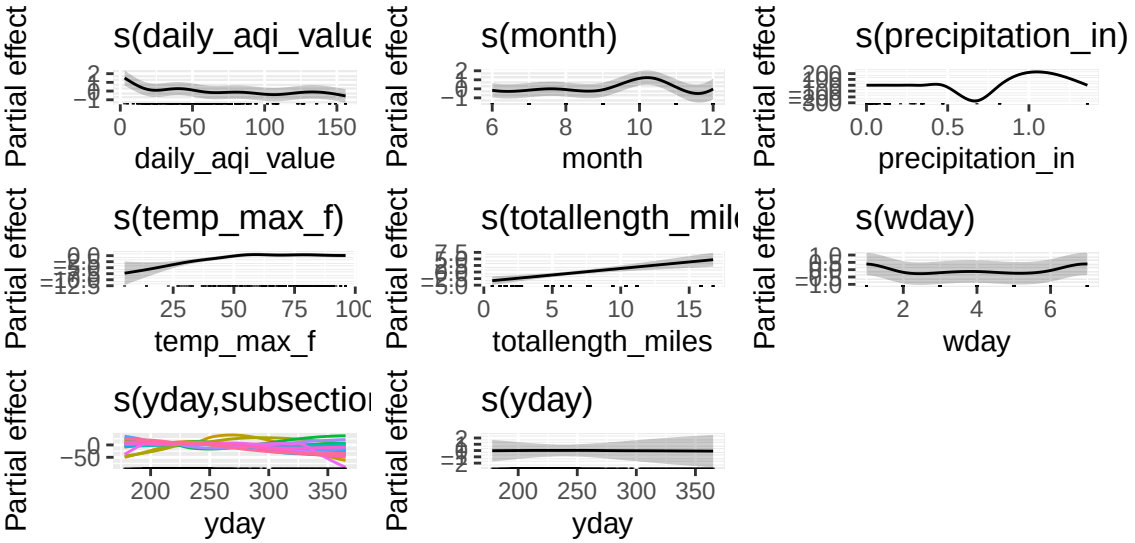


Figure 6.8: Global function (`s(yday)`) and group-specific deviations from the global function (`s(yday, subsection)`) for `gamm_modGS_ARMA_sub`).

```
#add the predicted values from the model
allTrail_GS_pred <- transform(allTrail_G,
                      mod_GS = predict(gamm_modGS_AR1$gam,
                                       type = "response"))

ggplot(allTrail_GS_pred, aes(x=mod_GS, y=max.camera)) +
  facet_wrap(~subsectionF, ncol= 3) +
  geom_point(alpha=0.3, aes(color = trailname)) +
  scale_color_manual(name = "Trail", values = colors) +
  geom_abline() +
  theme(legend.position="none") +
  labs(x="Predicted count", y="Observed count")
```

6.2.4 Single common smoother plus group-level smoothers with differing wiggliness (Model GI)

In this model class each group-specific smoother is permitted to have its own smoothing parameter and hence its own level of wiggliness. These models take the longest to run (as there are more smoothing parameters to estimate), but is useful if the different groups (here, trail subsections) differ in how 'wiggly' they are.

There are two major differences in how model GS was specified:

1. Explicit inclusion of a random effect for the intercept (the bs="re" term).
2. Specify m=1 instead of m=2 for the group-level smoothers. This allows for the marginal TPRS basis for relevent terms will penalize the squared first derivative of the function, rather than the second derivative. The aim is to reduce colinearity between the global smoother and the group-specific terms.

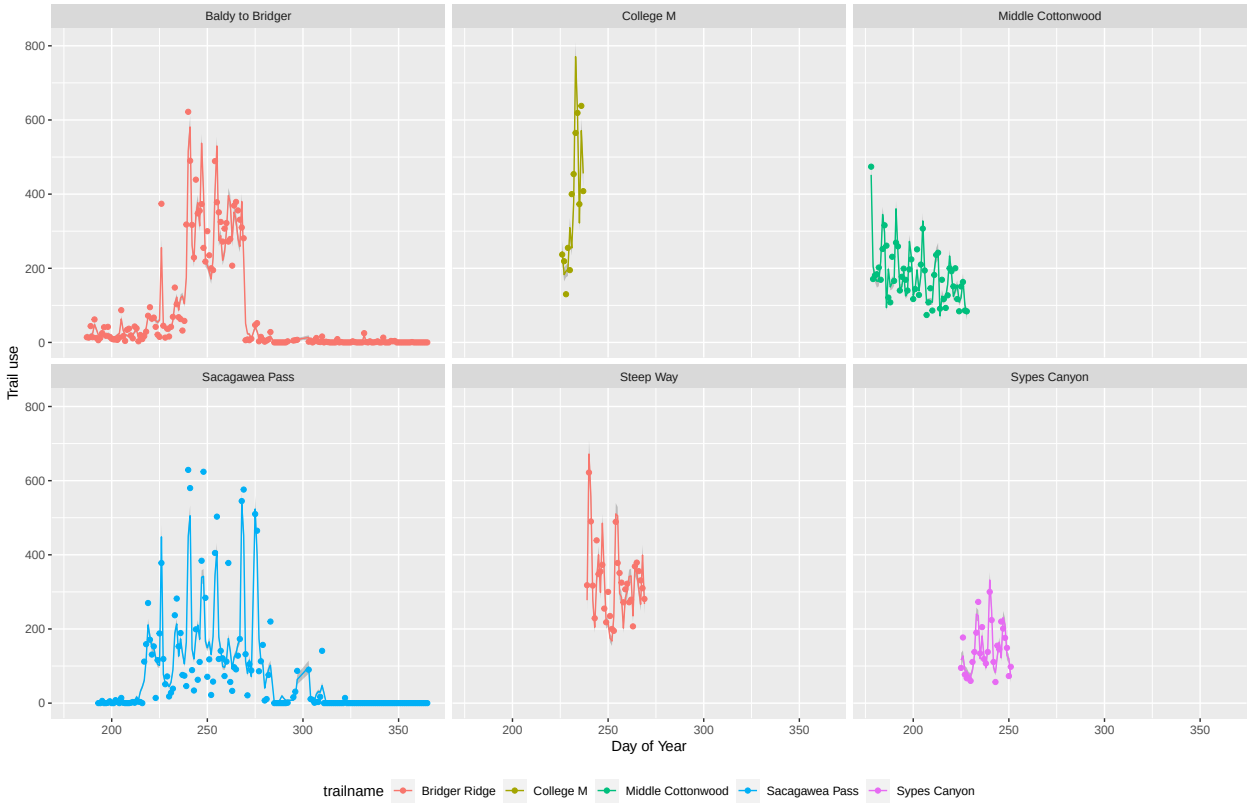


Figure 6.9: Predicted trail use count values (lines) versus observed trail use (points) for each high-use trail subsection, based on model *GS*.

Figure 6.10: Predicted trail use count values (lines) versus observed trail use (points) for each low-use trail subsection, based on model *GS*.

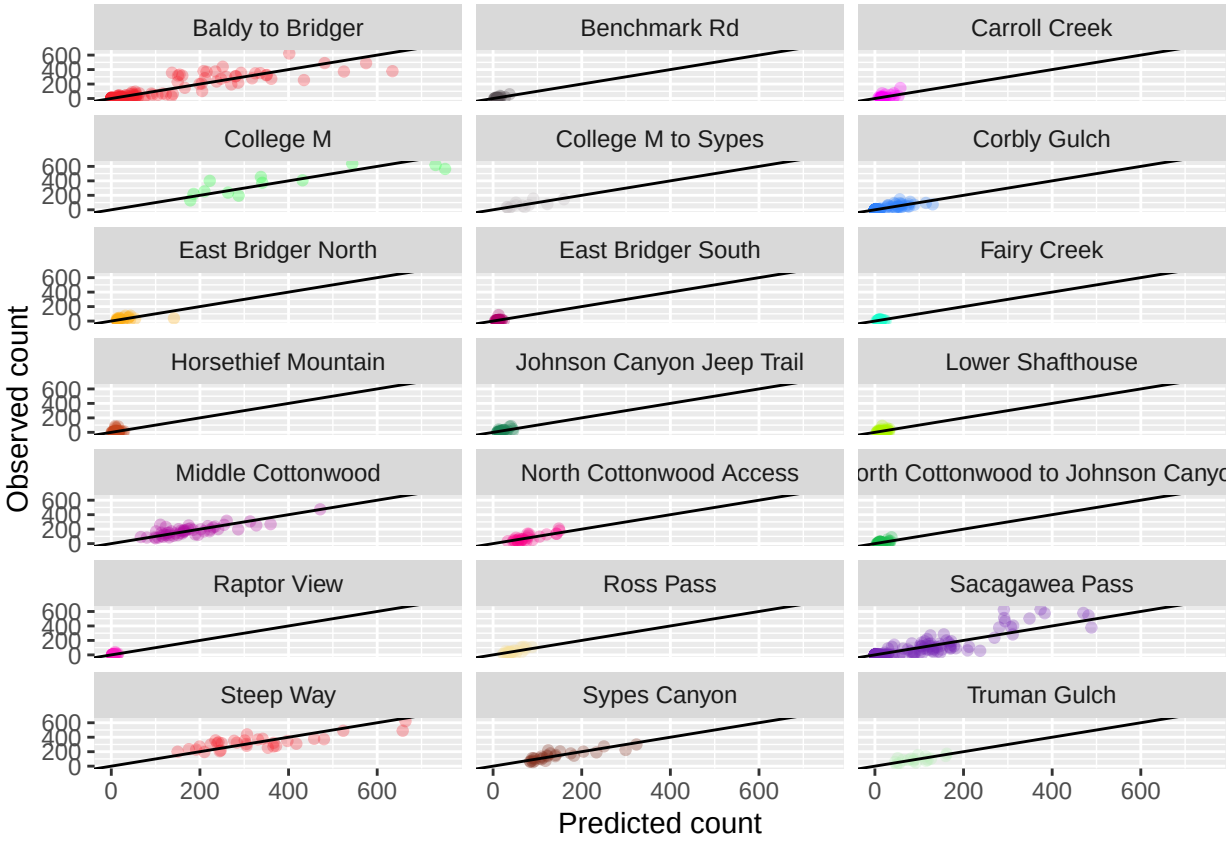


Figure 6.11: Assuming a well-fitted model GS, we would expect all trail subsections exhibiting similar patterns of dispersion around the 1-1 line (and as we are assuming the data is Poisson, the variance around the mean should equal the mean).

This approach in R looks like:

```
gamm_modGI_AR1 <- gamm(max.camera ~
                         s(yday, m=2, bs="cc") +
                         s(yday, by = subsectionF,
                           m=1, bs="cc") +
                         s(subsectionF,
                           bs="re",
                           # by = trailnameF,
                           k=21) +
                         # trailnameF +
                         s(month, bs = "cc",  k = 7) +
                         s(wday,
                           bs = "cc", k = 7) +
                         s(daily_aqi_value) +
                         s(temp_max_f) +
                         s(precipitation_in, k = 5) +
                        s(totallength_miles) +
                      total_traveltime +
                          max.count,
                        knots = list(yday = c(0,365),
                                     month = c(0, 13)),
                         correlation = corAR1(form = ~yday|subsectionF),
                       data = allTrail,
                       family = poisson)
```

When including a by-variable smooth, you are allowing for different smoothness parameters for each level of `subsectionF`.

Here, we present the summary and diagnostic plots for model GI with an AR1 temporal structure.

```
##                     Model df      AIC       BIC    logLik   Test  L.Ratio p-value
## gamm_modGI$lme          1 36 126685.2 126864.08 -63306.58
## gamm_modGI_AR1$lme      2 37  66969.6  67153.49 -33447.80 1 vs 2 59717.57  <.0001
```

The summary now includes all the different day of year (`yday`) by trail subsection (`subsectionF`) parameters.

```
##
## Family: poisson
## Link function: log
##
## Formula:
## max.camera ~ s(yday, m = 2, bs = "cc") + s(yday, by = subsectionF,
##     m = 1, bs = "cc") + s(subsectionF, bs = "re", k = sub.number) +
##     s(month, bs = "cc", k = 7) + s(wday, bs = "cc", k = 7) +
##     s(daily_aqi_value) + s(temp_max_f) + s(precipitation_in,
##     k = 5) + s(totallength_miles) + total_traveltime + max.count
##
## Parametric coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.8992494  0.3759070   2.392   0.0169 *
## total_traveltime -0.0401890  0.0082482  -4.872 1.29e-06 ***
## max.count         0.0116742  0.0005485  21.282  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                                                       edf Ref.df        F  p-value
## s(yday)                                          7.127e+00  8.000  994.511  < 2e-16 ***
## s(yday):subsectionFBaldy to Bridger              6.162e+00  8.000 1195.664  < 2e-16 ***
## s(yday):subsectionFBenchmark Rd                  2.175e+00  5.000  358.320 0.049930 *
## s(yday):subsectionFCarroll Creek                 1.416e+00  5.000    3.342 0.787660
## s(yday):subsectionFCollege M                     3.800e+00  5.000 1078.270 1.86e-05 ***
## s(yday):subsectionFCollege M to Sypes            3.563e+00  5.000  537.253 7.88e-07 ***
## s(yday):subsectionFCorbly Gulch                  7.504e+00  8.000 1062.789  < 2e-16 ***
## s(yday):subsectionFEast Bridger North            1.237e-04  6.000    0.000 0.416870
## s(yday):subsectionFEast Bridger South            3.047e+00  6.000  872.823 6.56e-05 ***
## s(yday):subsectionFFairy Creek                   3.310e+00  5.000 1191.535 0.000547 ***
## s(yday):subsectionFHorsethief Mountain           8.804e-05  6.000    0.000 0.454800
## s(yday):subsectionFJohnson Canyon Jeep Trail     2.233e+00  5.000  418.336 0.005384 **
## s(yday):subsectionFLower Shafthouse              2.883e+00  7.000  185.441 0.001892 **
## s(yday):subsectionFMiddle Cottonwood             4.154e+00  5.000 2127.067 5.46e-05 ***
## s(yday):subsectionFNorth Cottonwood Access       4.338e+00  5.000 1498.815 0.001054 **
## s(yday):subsectionFNorth Cottonwood to Johnson Canyon 4.663e+00  6.000 1049.535 3.97e-06 ***
## s(yday):subsectionFRaptor View                   1.281e+00  5.000  160.742 0.053252 .
## s(yday):subsectionFRoss Pass                     1.198e+00  4.000  211.833 4.21e-05 ***
## s(yday):subsectionFSacagawea Pass                7.971e+00  8.000 3361.353  < 2e-16 ***
## s(yday):subsectionFSteep Way                     4.624e+00  5.000  122.084 0.004566 **
## s(yday):subsectionFSypes Canyon                  1.308e-05  5.000    0.000 0.391803
## s(yday):subsectionFTruman Gulch                  3.688e+00  5.000  651.198 0.009058 **
## s(subsectionF)                                   4.968e+00 19.000    4.635  < 2e-16 ***
## s(month)                                         3.623e+00  5.000   11.556  < 2e-16 ***
## s(wday)                                          4.976e+00  5.000  461.824  < 2e-16 ***
## s(daily_aqi_value)                               8.912e+00  8.912  170.707  < 2e-16 ***
## s(temp_max_f)                                    8.724e+00  8.724 1039.786  < 2e-16 ***
## s(precipitation_in)                              3.744e+00  3.744  103.796  < 2e-16 ***
## s(totallength_miles)                             1.000e+00  1.000   39.097  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.843
##   Scale est. = 1         n = 1064
```

Figure 6.12 shows a subsample of the group-specific smoothers from this model. Some trail subsections (e.g., East Bridger South) have very similar shapes to the global smoother. Others do differ from this trend with lower trail use earlier in the year and higher in later times (e.g., Baldy to Bridger) or a mix of variations from the global trend (e.g., Corbly Gulch, Sacagawea Pass, Steep Way).

Figures 6.13 and 6.14 shows that model GI is the first and only model to fully capture the temporal autocorrelation (through the AR1 structure). There is only slight evidence for overdispersion (Figures **??** and 6.16) which seems most likely due to a slightly zero-inflated distribution.

```
rg <- gratia::rootogram(gamm_modGI_AR1$gam)
draw(rg)
```

```
#add the predicted values from the model
allTrail_GI_pred <- transform(allTrail_G,
                      mod_GI = predict(gamm_modGI_AR1$gam,
```

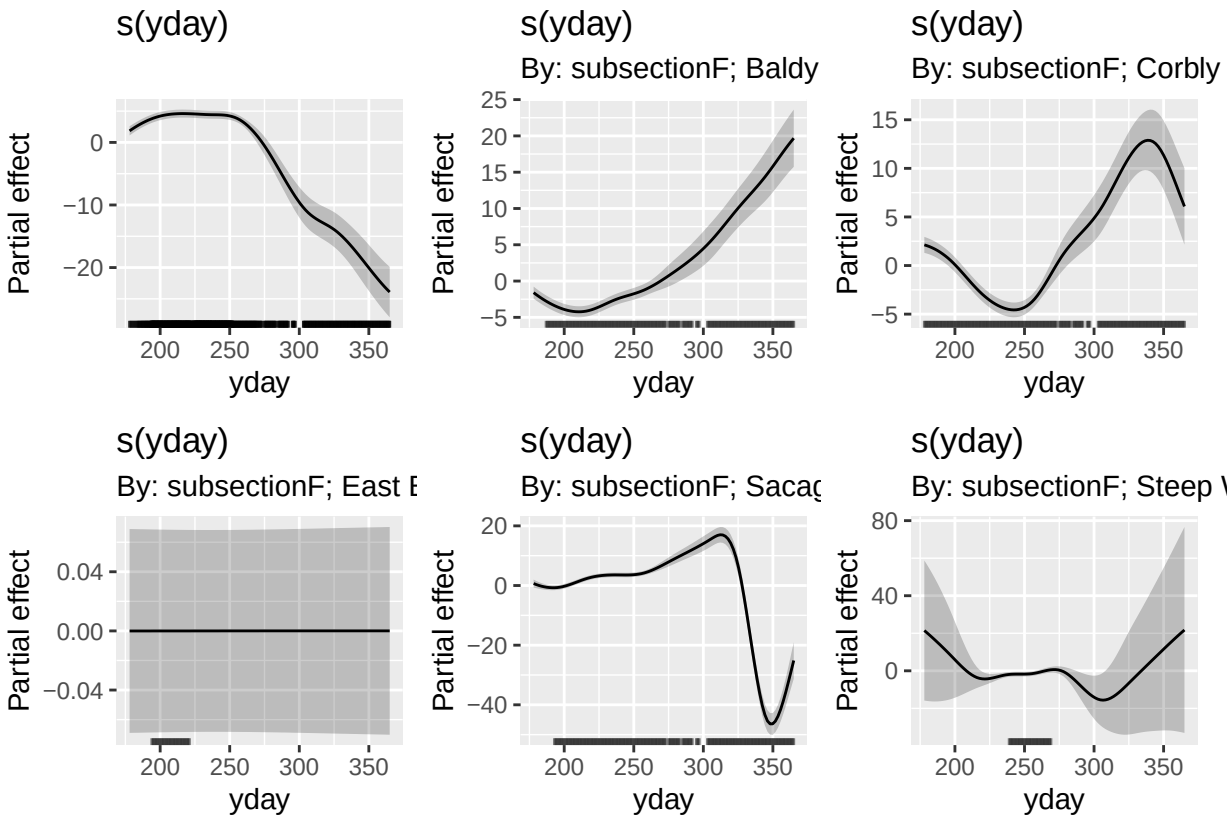


Figure 6.12: Subsection of partial effect plots for model *GI*.

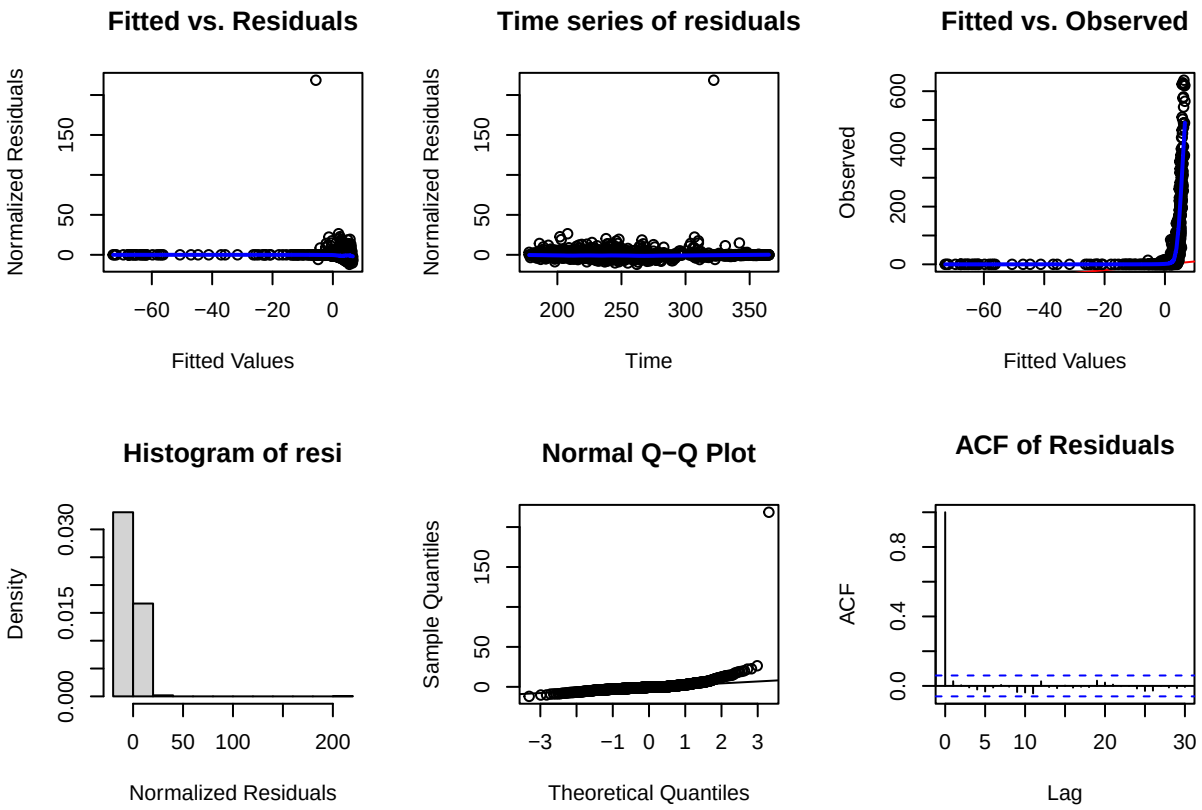


Figure 6.13: Times series diagnosis plots for model *GI*.

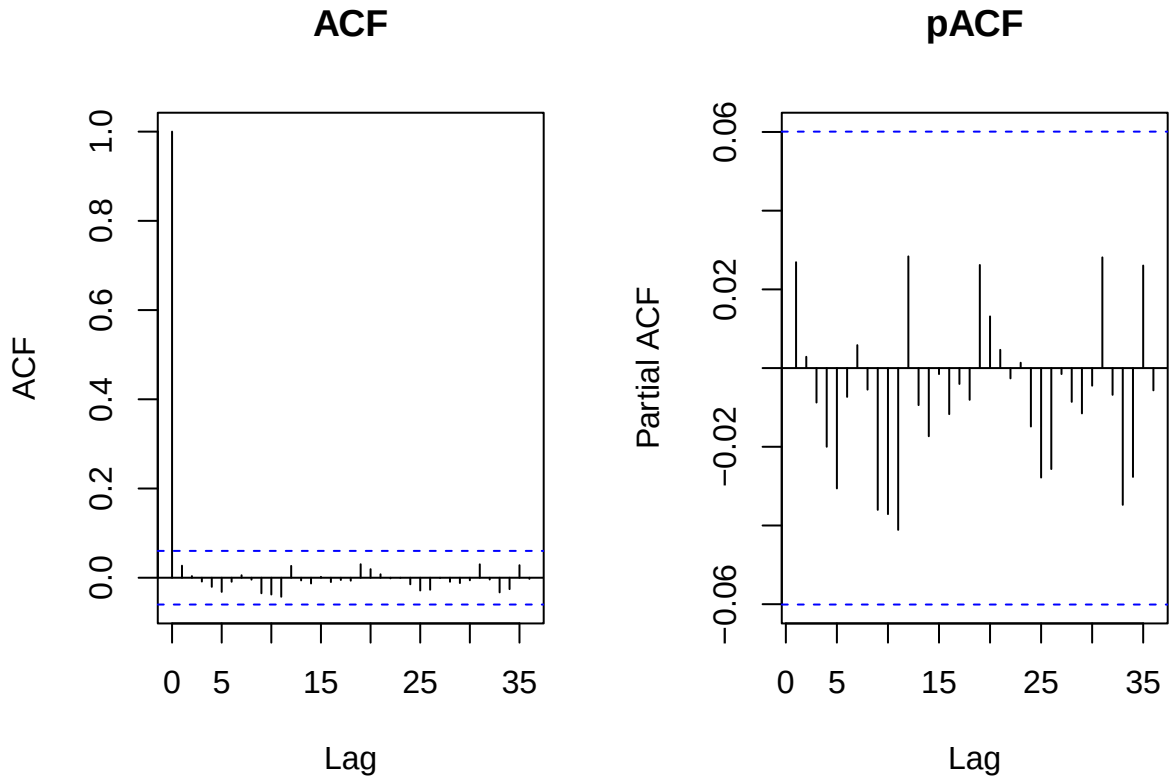


Figure 6.14: Times series ACF/pACF diagnosis plots for model *GI*.

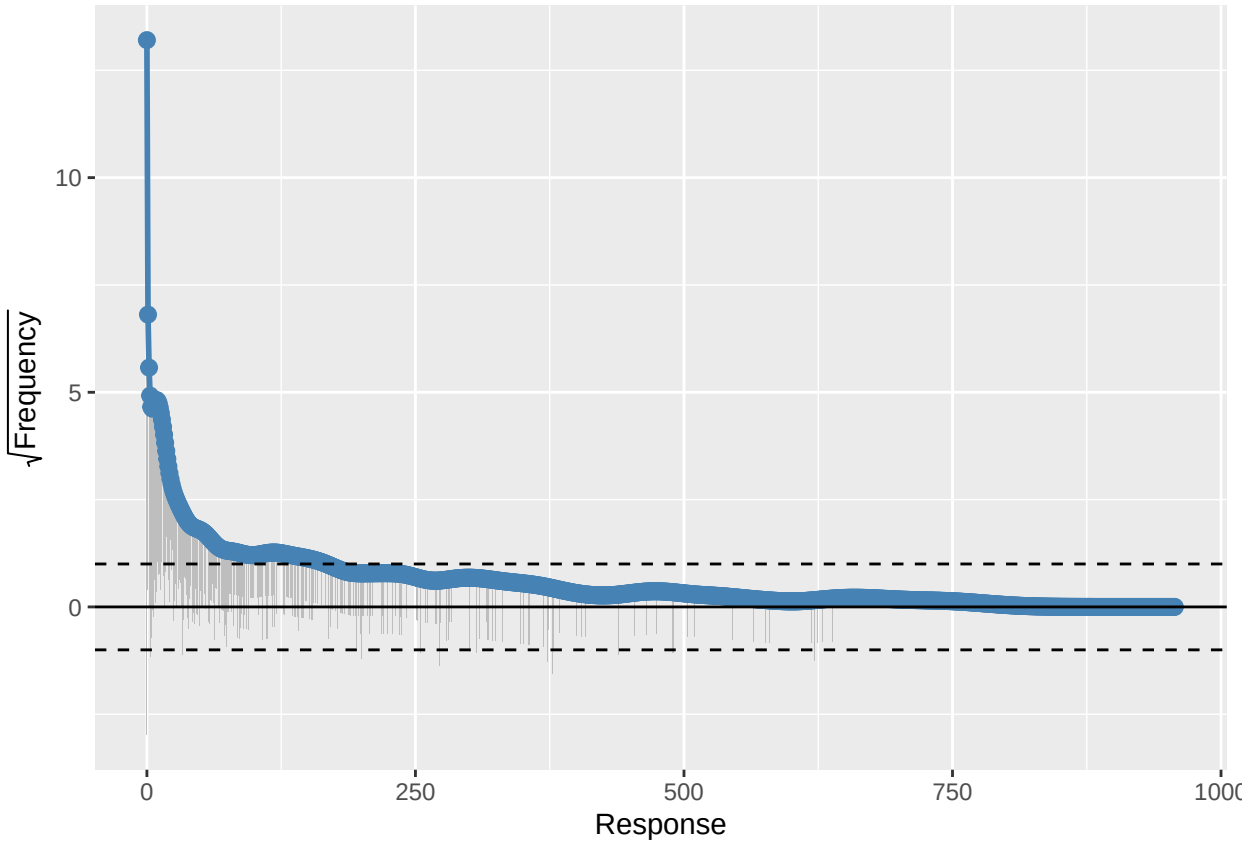


Figure 6.15: Rootogram for checking for overdispersion.

```
                                    type = "response"))

ggplot(allTrail_GI_pred, aes(x=mod_GI, y=max.camera)) +
  facet_wrap(~subsectionF, ncol= 3) +
  geom_point(alpha=0.3, aes(color = trailname)) +
  scale_color_manual(name = "Trail", values = colors) +
  geom_abline() +
  theme(legend.position="none") +
  labs(x="Predicted count", y="Observed count")
```

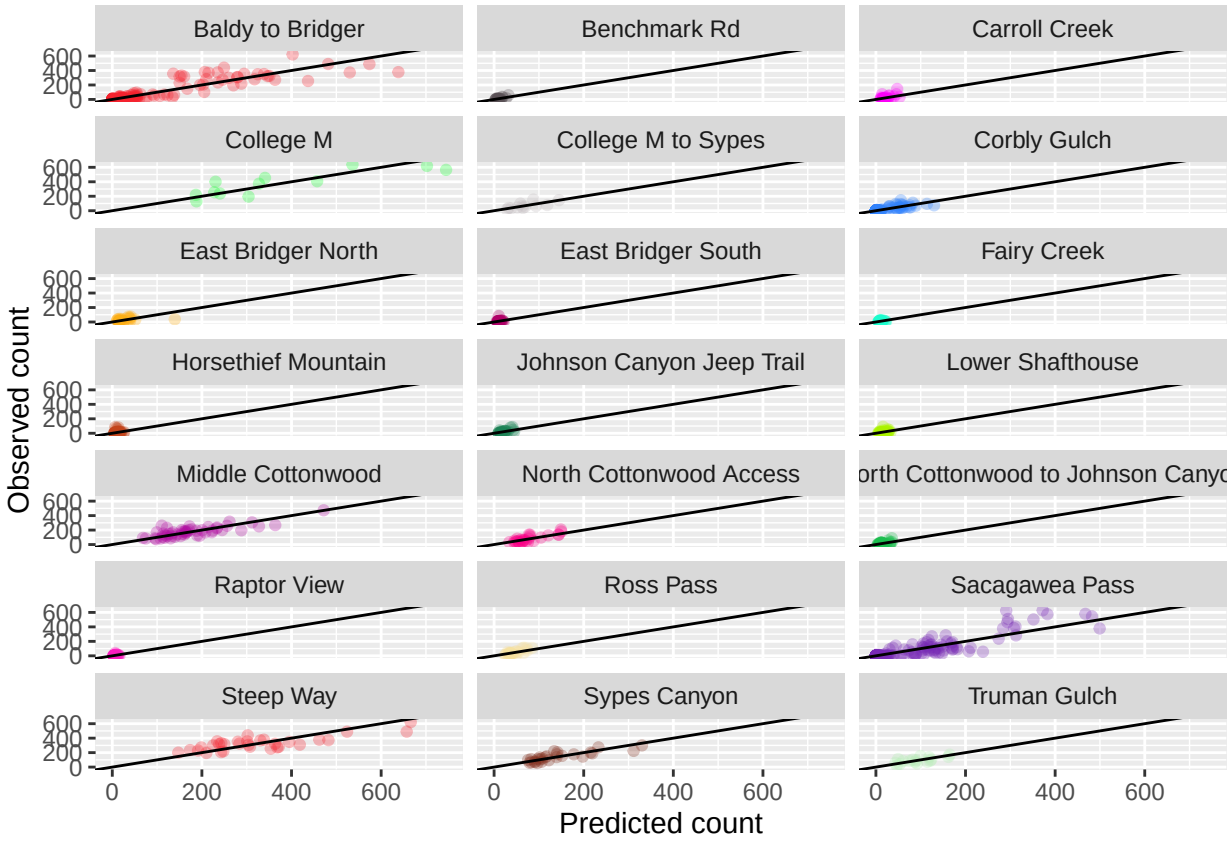


Figure 6.16: Assuming a well-fitted model GI, we would expect all trail subsections exhibiting similar patterns of dispersion around the 1-1 line (and as we are assuming the data is Poisson, the variance around the mean should equal the mean).

Figures 6.17 and 6.18 show the predictive values and observed data points for model GI.

6.3 Prediction/Forecasting

Up until this point we have looked at predictions for in-sample data. Effectively interpolating with our predictions. To examine the usefulness of these models for forecasting (i.e. predicting trail use in the near future) we may look at how these models handle temporal extrapolation (i.e. prediction outside of the range of data used to fit the models). We discussed in Section 5.4 that extrapolation in the GA(M)M framework can be tricky due to how the model uses splines to learn from the data via the basis functions.

Figures 6.19 and **??** both show predicted trail use for the best model within each G, GS, and GI for the entire year of 2021. Observed trail use counts are plotted (colored by day of week). Both Models GS and GI show

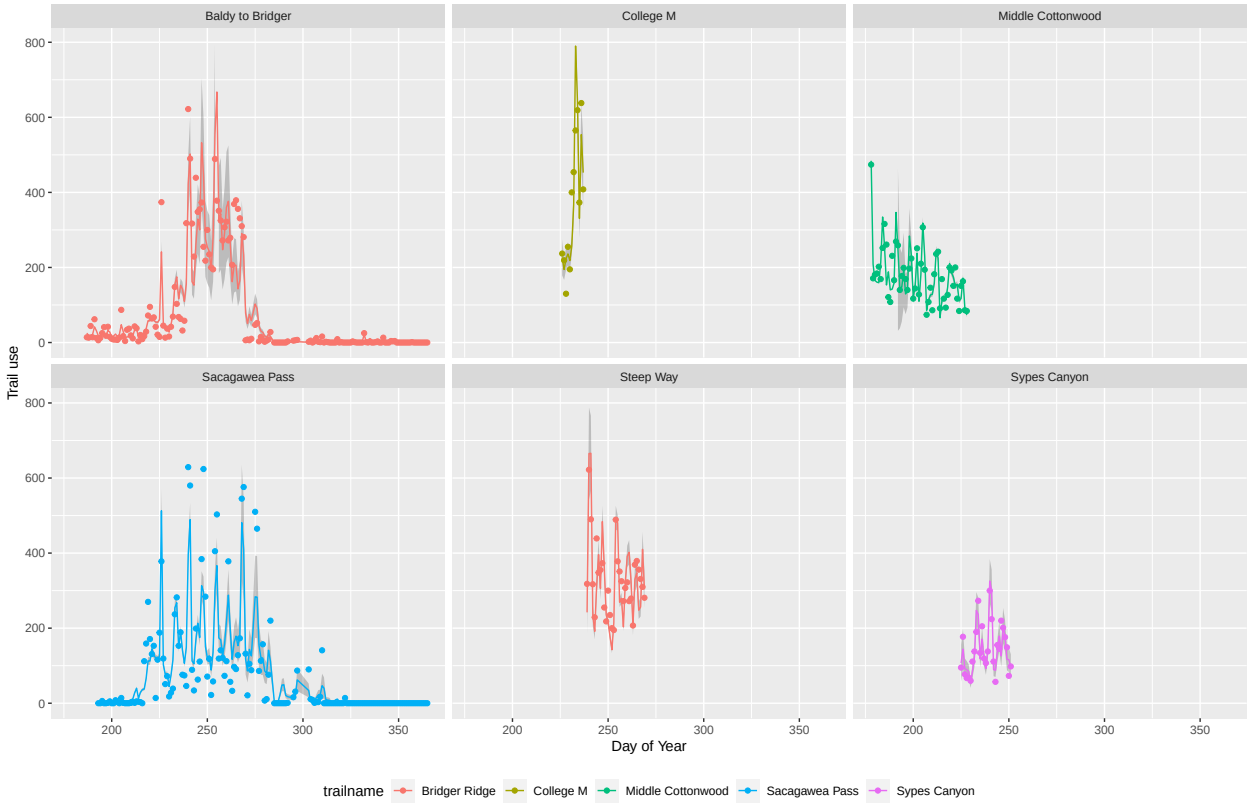


Figure 6.17: Predicted trail use count values (lines) versus observed trail use (points) for each high-use trail subsection, based on model *GI*.

Figure 6.18: Predicted trail use count values (lines) versus observed trail use (points) for each low-use trail subsection, based on model *GI*.

improvement over Model G, which is to be expected due to increase variability between trail subsections. Some trail subsections (e.g. Steep Way) still show unreasonably high predictions (and large error ribbons), however trails with a longer duration of observations (e.g. Baldy to Bridger) show improvement. Several model specification choices have improved these predictions over past iterations of models (not shown). For example, the use of cyclic cubic splines for time predictor variables (`bs= "cc"`) allows information from winter observations late in the year to inform predictions for early in the year. Also, restricting the number of knots for `precipitation_in` (now $k = 5$ when before it was allowed to default to $k = 10$) has removed some odd jumps in predictions on days with high precipitation. One notable difficulty for these models across all trails is the ability to predict higher than typical trail use days.

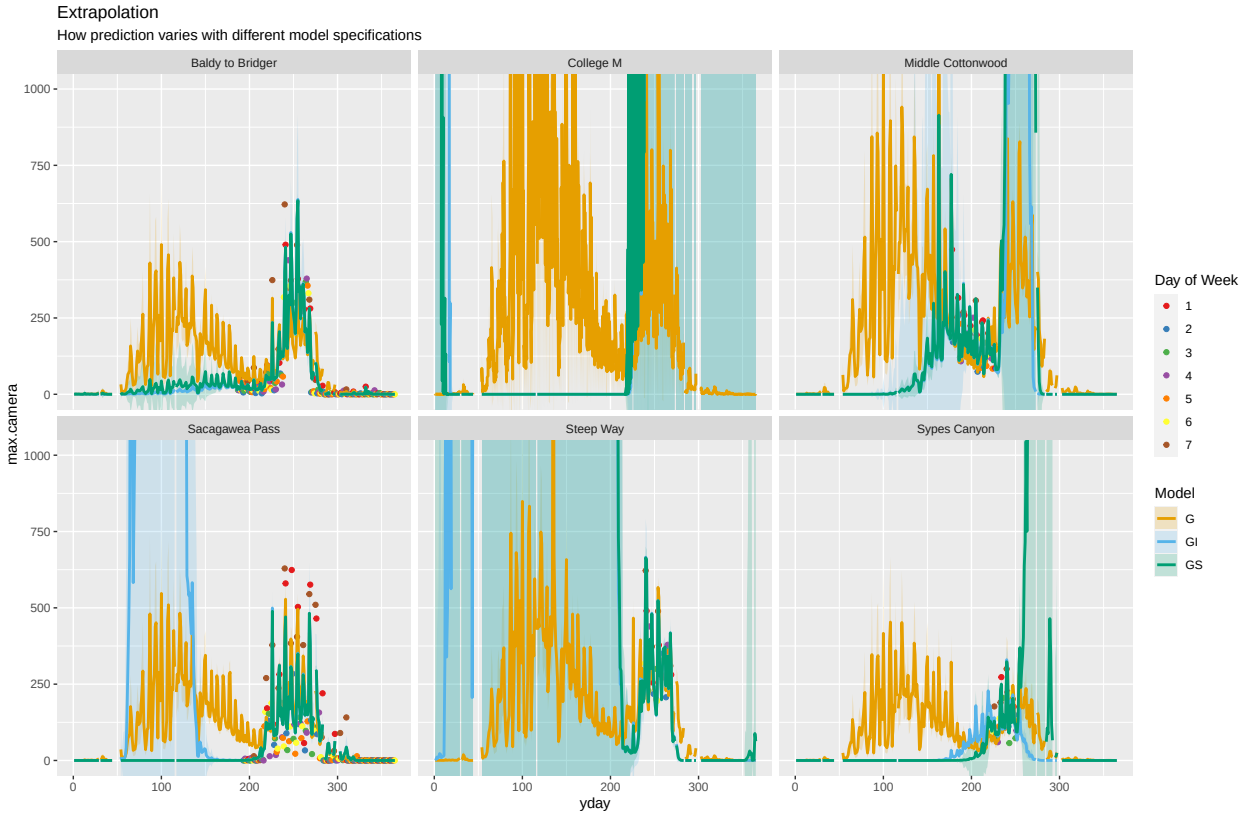


Figure 6.19: Predicted trail use count values (lines) versus observed trail use (points) for each high-use trail subsection over an entire year, based on each model (G, GS, and GI). Observed trip counts (data points) are colored by day of the week and show the higher use on weekends.

To be added: comparison of Middle Cottonwood predicitons for single trail model vs this joint model. Currently the single trail model also uses AllTrails data which has duplicate search view numbers for certain days. This messes with my code that combines all predictions for plotting. When fixed the plots will show that models that share data across subtrails improve predictions over models with individual trail data.

Figure 6.21 shows predictions from the top model within each category for trails in Bridger Mountains without deployed camera counters. All other covariate data is available for the year 2021.

To be added (hopefully) animated plots of prediction values and residuals plotted spatially. Currently these animations are super cursed and are not yet up and running.

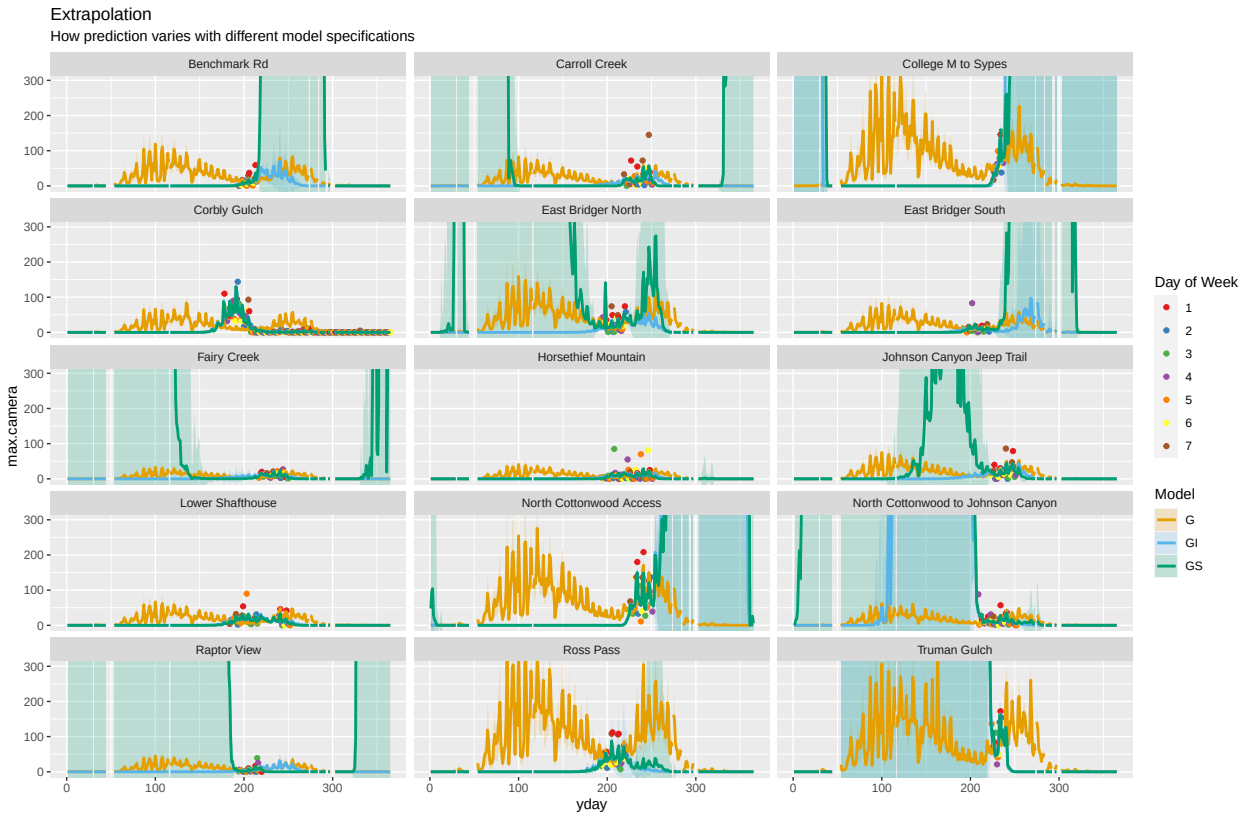


Figure 6.20: Predicted trail use count values (lines) versus observed trail use (points) for each low-use trail subsection, based on each model (*G*, *GS*, and *GI*).

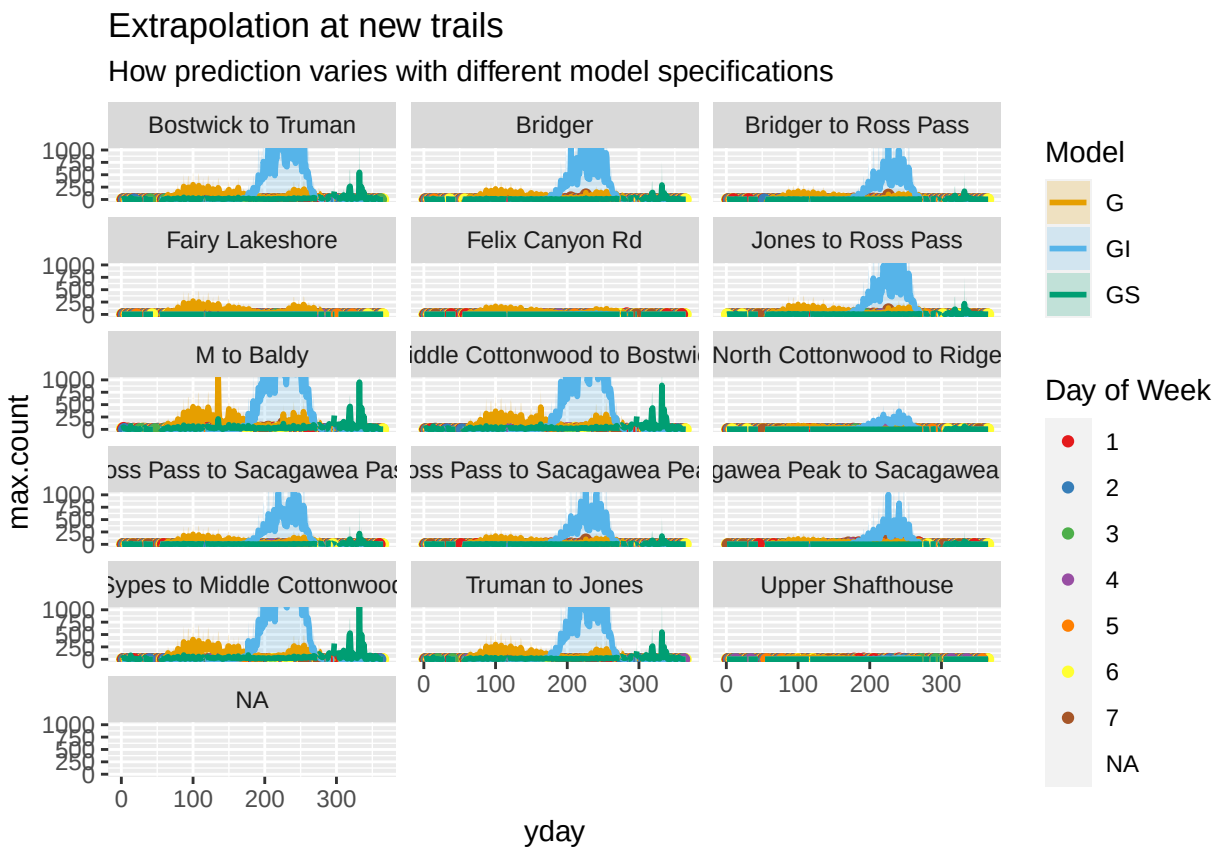


Figure 6.21: Predicitons at Bridger Mountain trails without deployed camera counters.

Table 6.1: AIC table comparing model fits

Model	df	AIC	deltaAIC
gamm_modG_ARMA$lme	20	31217	0
gamm_modGS_AR1$lme	17	68829	37612
gamm_modGI_AR1$lme	37	66970	35752

6.4 Results

Comparing models based on AIC is a robust approach to comparing the different model structures. It has been shown to be an appropriate means of model comparison for models fit with `gam()`, but has not been similarly confirmed for models fit with `gamm()`. While we use this approach here, results should be taken with a grain of salt. For example, even though visual inspection of predictions of models GS and GI seem to show evidence of improvement over model G, Table 6.1 does not provides evidence for including among-group functional variability. In Pedersen et al. (2019), they caution against selecting models based purely on AIC. Instead, model selection should be based on expert subject knowledge about the system, computational time, and most importantly, the inferential goals of the study.

Our goal to select the model that has the best predictive ability across various trail types. This can be checked by holding some fraction of the data out (e.g., a single trail subsection) prior to the analysis and comparing how well different models fit that data. To this end, we have left out the New World Gulch trail and fit our selected model for each type (G, GS, GI) to these out-of-sample data. To evaluate how well each model fits this new data, we calculated the total deviance of the out-of-sample data. The deviance is equal to two times the sum of the difference between the log-likelihood of the out-of-sample data (as predicted by each model) and a saturated model that has one predictor for each data point, all multiplied by the scale parameter for the family of interest. It can be interpreted similarly to the residual sum of squares for a simple linear regression (Wood, 2017a, p. 109).

To be added: get predictions for New World Gulch (currently missing trail characteristic data which is needed).

Table 6.2 shows that both models GS and GI are better at predicting in-sample fits for all trail subsections in this analysis. While this provides additional evidence for including inter-group variability in our model, it does contradict our AIC findings.

6.5 Conclusions

The GAM framework and its extensions that allow for modelling non-linear and linear relationships between response variables and predictors with random effects and a way to account for temporal autocorrelation is a valuable tool for a statistical analysis of trail use in the Bridger Mountains. Limitations of this model includes not being able to incorporate both spatial and temporal autocorrelation without some heavy lifting required. It's not clear how to model nested random effects of trail and trail subsections in this model. Section B provides a look at an alternative model that allows for spatial correlation to be incorporated instead of temporal.

While the GAM framework for modelling is a powerful tool, it also requires a lot of modeling specification decisions. In this report we have discussed different options for model type that allows for different ways to provide a global smoother and inter-trail variation (models G, GS, and GI), the choice of basis functions and number of knots, differing ways to account for temporal correlation, and what/how many predictor variables to include. Model complexity could increase if interaction terms were to be included as well. Another complication is that while tools for diagnotics and model comparison are well developed for models fit with the `gam()` function, when we fit with the `gamm()` function (as we do in this model with our timeseries data) extra care must be taken as not all tools are appropriate in this case. Packages such as the **gratia**

Table 6.2: Predictive ability for models *G*, *GS*, and *GI* applied to the Bridger Mountain trail use dataset. Deviance values represent the total deviance of model predictions from observations. Intercept only results are for a null model with only subsection-level random effect intercepts included.

	Total deviance			
subsectionF	Intercept only	Model G	Model GS	Model GI
Baldy to Bridger	29338	6851	3245	3253
Benchmark Rd	287	221	116	130
Carroll Creek	852	600	533	571
College M	886	354	306	297
College M to Sypes	280	154	129	112
Corbly Gulch	6241	6068	731	730
East Bridger North	449	421	430	437
East Bridger South	322	348	300	313
Fairy Creek	130	226	142	148
Horsethief Mountain	1161	1261	1117	1156
Johnson Canyon Jeep Trail	527	311	291	291
Lower Shafthouse	600	875	481	484
Middle Cottonwood	1396	1175	591	601
North Cottonwood Access	807	378	279	278
North Cottonwood to Johnson Canyon	615	727	375	377
Raptor View	232	247	166	214
Ross Pass	535	294	224	224
Sacagawea Pass	29194	7959	4674	4648
Steep Way	748	651	533	542
Sypes Canyon	744	352	213	231
Truman Gulch	300	385	186	184

package which provides visulizing and diagnostic functions for GAMs is still adding functionality and will likely provide additional tools for future analyses. Even with all of this considered, this modelling framework has been shown to be a promising way to analyse recreational trail use and contains a lot of flexibility while being able to not only provide predictions at new trails/times of year but also valuable insight into trends for each predictor variable. The predictive abilities of these models will increase as additional data is made available.

With this in mind, we provide the following recommendations for future observations periods and analyses.

1. It is better to have cameras deployed for longer periods of times. Year-round data for even a subset of trails will help improve predictions across all trails.
2. Subsections within a trail seem to capture similar data. Cameras spread across more trails may provide more information than multiple camera counters deployed along a single trail.
3. Multiple years of data would provide valuable information about annual trends and be helpful in forecasting trail use into the future.
4. If spatiotemporal correlation structure is available for future analysis then it would be helpful to consider the spatial network node/edge designations in tandem with camera placement.

Section 7

Trade-Offs in Prediction Accuracy

A secondary aim of this report is to assess tradeoffs in predictive accuracy for the statistical application. Here, we investigate several scenarios where we anticipate discrepancies in predictive abilities.

7.1 High Use versus Low Use

Due to variation in trail use across the network of trails in the Bridger Mountains we anticipate different levels of predictive ability between trails of high and low use. To provide a comparison for predictive accuracy, we first assign trails (at the subsection level) to be a "high" or "low" use trail based on expert input by Headwaters Economics. The following categories were determined:

High Use: Baldy to Bridger, Bridger, Bridger to Ross Pass, College M, M to Baldy, Middle Cottonwood, Sypes Canyon, Ross Pass to Sacagawea Peak, Sacagawea Pass, Steep Way

Low Use: Fairy Creek, Horsethief Mountain, Carroll Creek, Raptor View, College M to Sypes, Truman Gulch, East Bridger South, East Bridger North, Lower Shafthouse, Corbly Gulch, North Cottonwood to Johnson Canyon, North Cottonwood Access, Johnson Canyon Jeep Trail, Benchmark Rd

We use deviance as our chosen metric for assessing predictive accuracy. We looked at models fit with types G, GS, and GI (see Section 6 for details) and then obtained predictions for those same (in-sample) trail subsections. To account for different number of days of observation between these two groups the calculated deviance for each model was divided by the number of days of observations to find an average deviance measure. Table @ref(tab:deviance_highlow_kable) shows that Model GS provided the best fit for both two groupings of trials. Additionally the deviance measure is lowest for the "low" use trial group. One explanation is that all models fit to these data had a difficult time predicting on days with higher than typical trail use. These events are more likely to occur on high use trails and thus the predictions for these trails will have a higher deviance overall.

	Total deviance			
highlow	Intercept.only	Model.G	Model.GS	Model.GI
high	136.93	38.11	21.02	21.04
low	21.90	20.55	9.03	9.28

	Total deviance			
Motor Vehicle Use	Intercept Only	Model G	Model GS	Model GI
Dirt Bikes (seasonal)	30.81	28.81	6.29	6.32
Non-Motorized	97.14	30.58	18.20	18.34
Wheeled OHV 50" or <	15.48	11.71	9.33	9.83

	Total deviance			
Parking Lot Size	Intercept Only	Model G	Model GS	Model GI
L	110.09	32.34	18.66	18.70
M	20.83	16.98	10.64	11.06
S	28.66	27.33	7.62	7.80

7.2 Places with different types of use

We also want to investigate how prediction accuracy differs in areas of varying trail use (e.g., do motorized trails differ in important ways from non-motorized trails?).

7.2.1 Motor Vehicle Use

7.2.2 Parking Lot Size

3. An analysis of the change in predictive accuracy as the number of trail counters used changes.

- The GAMM model applied does not allow for multiple measures per unit time. In an ideal world trail counters would be deployed at as many different trails for as long as possible. Longer deployment times would help to model season and annual trends. It is possible that with more temporal data coverage we could use simpler models (i.e. a gam or gamm approach without the need for temporal autocorrelation structure) that would save on computation time).
- Need to talk about forecasting (or temporal extrapolation). If we want to forecast in the future for a specific date or time frame we really need prior observations on those intervals (i.e. previous years). This is not unique to GAM models, predicting beyond the range of observed samples is tough.

*Can we identify trails that are “closest” to global smooth trend as ideal condidates for year-round cameras?

Appendix A

Utility of All Trails auxilary data

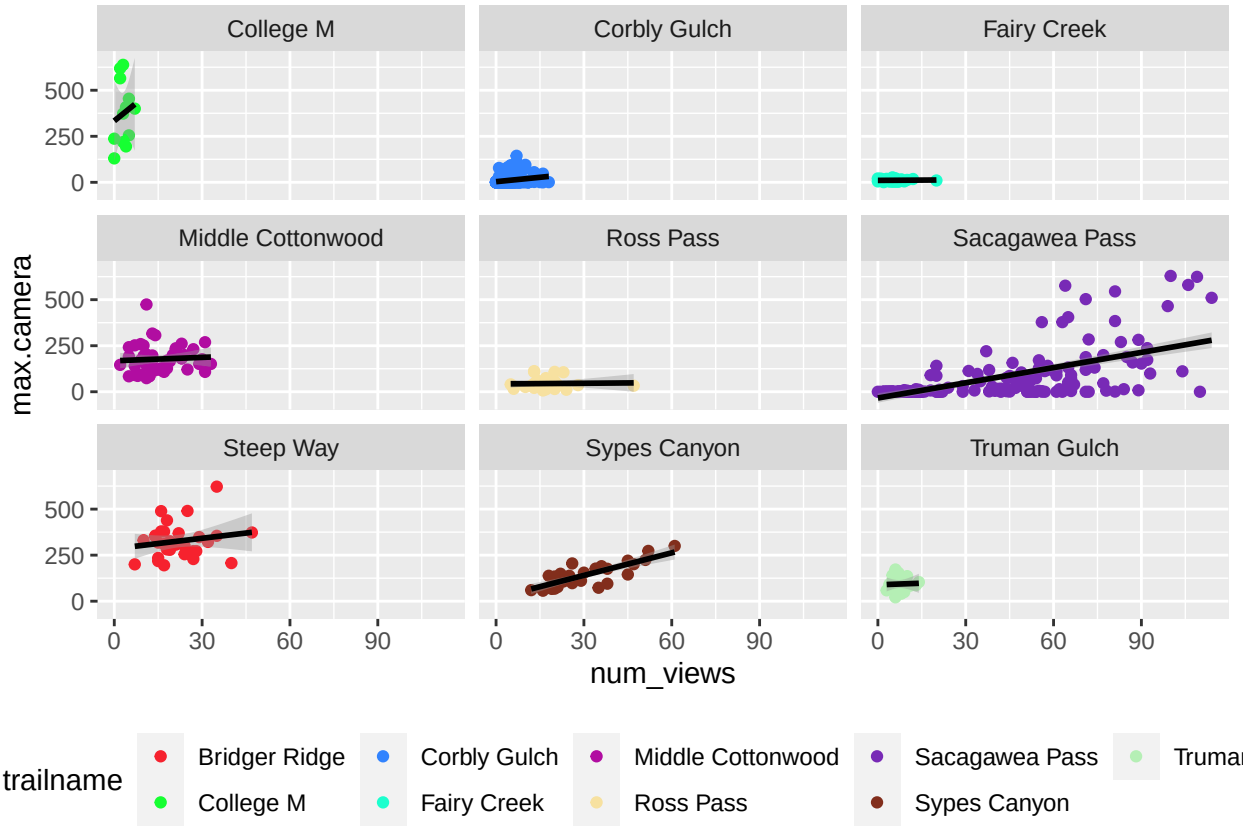
Through a partnership between AllTrails and Headwaters Economics, AllTrails has provided search data for a subset of trails in the Bridger Mountains to be used as auxillary data for the Bridger Mountain trail use statistical analysis. The following trails are included in this dataset:

1. Fairy Creek
2. College M
3. Bridger Ridge
4. Sacagawea Pass
5. Sypes Canyon
6. Truman Gulch
7. Corbly Gulch
8. Ross Pass
9. Middle Cottonwood

An overview of the available AllTrials data is presented in Section 3.1.1.3. In this report we aim to investigate the utility of these data (and potential transformation of the data) as a predictor variable in the trail use analysis. We are interested in whether some transformation of the data (e.g., moving average of views or perhaps the cumulative sum of views over the preceding n days) may prove more informative than the raw data themselves.

A.1 Data Visualization

We start with exploring some visualizations of different data transformations. Figure @ref{fig:AT-vis-views} shows potential linear relationships between the different trail subsections and the untransformed number of views for each. While not all trail subsections exhibit evidence for a linear relationship (or non-linear) we do see evidence for a positive relationship with Sacagawea Pass and Sypes Way. Additional data (days of year or different trail subsections) could still hold more evidence for this relationship.



We started our exploration of potential data transformations with a 7-day moving average that takes the preceding 7 days to average over. This assumes the recreational users might be conducting trail searches in the week leading up to a trail use event. Figure A.1 shows the relationship between trail use by camera counter and AllTrails searches as this 7 day moving average. There is not any increase in the number or strength of linear relationships for any of the trail subsections. Sypes Canyon no longer has quite as prominent of a positive linear relationship between the two variables. Using a shorter time frame for the moving average does not seem to provide any improvement (not shown).

We also explored using the cumulative sum of the previous n days. Figure **??** shows that this approach can show a different linear relationship compared to non-transformed search views. Both Steep Way and Middle Cottonwood now show evidence of *negative* linear relationships between camera counts and search views.

What these visualizations show is that we must be careful in our choice for how we incorporate these data into our model for trail use.

A.2 Fitting Models

Another consideration is how to include AllTrails search data in our GAM framework with (or instead of) the Strava trip count data. We fit and compare the following models:

1. Model GI with AR1 temporal correlation structure and only the Alltrail search views as the raw data.
2. Model GI-AR1 with only the AllTrails search data as a 7 day moving average.
3. Model GI-AR1 with both the Strava trip counts and the Alltrail search views as the raw data.
4. Model GI-AR1 with both the Strava trip counts and the Alltrail search views as a 3 day moving average.

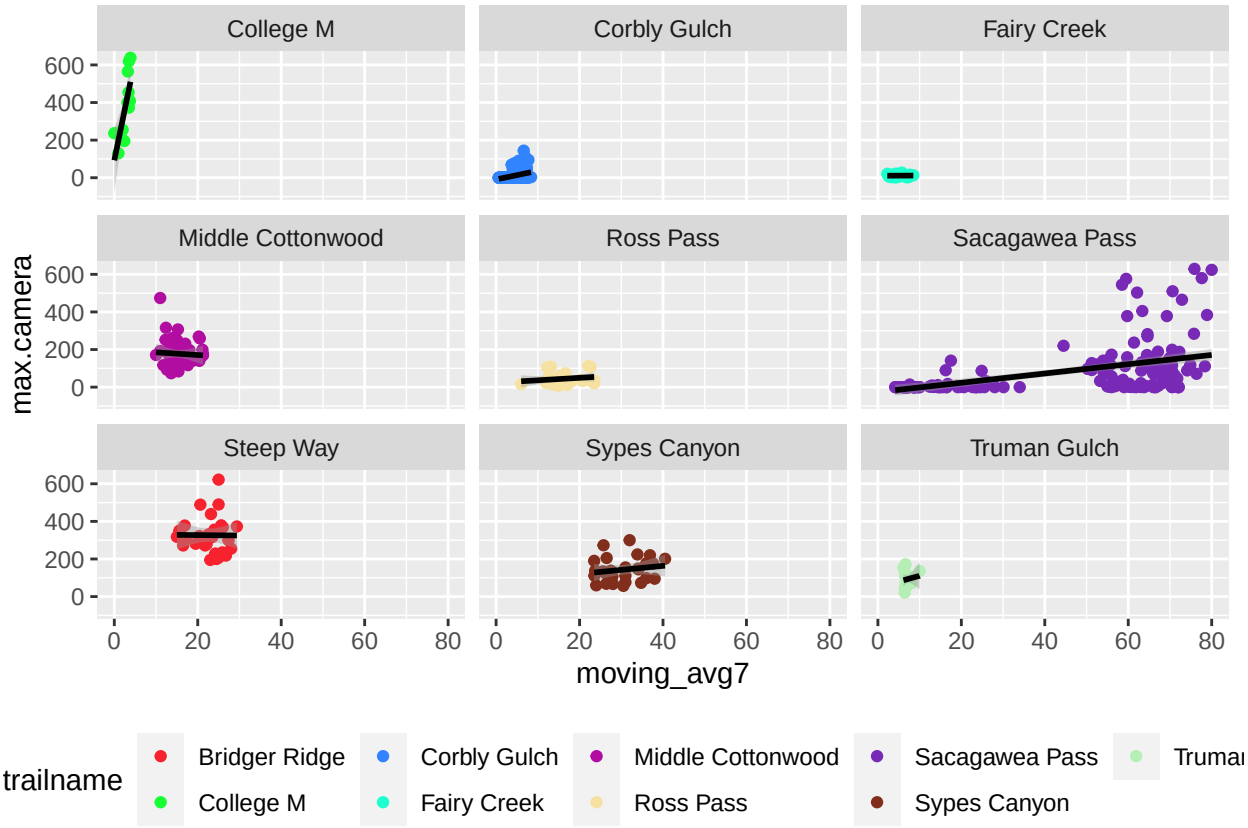


Figure A.1: Scatterplot of trail use camera counts and AllTrails search views represented as a moving average of the number of views for the preceding seven days. Linear relationship fit (line) plotted in black.

Table A.1: Predictive ability for models examining inclusion of AllTrails and Strava Metro data applied to the subsection of Bridger Mountain trail use dataset where both predictor variables are available.

	Total deviance		
subsectionF	Model GI-AT-Raw	Model GI-Both-MA3	Model GI-Strava
College M	205	251	305
Corbly Gulch	756	719	721
Fairy Creek	135	143	140
Middle Cottonwood	627	547	623
Ross Pass	209	263	219
Sacagawea Pass	3485	3565	4294
Steep Way	964	624	589
Sypes Canyon	248	182	200
Truman Gulch	225	216	179

A.3 Comparing Models

We compared models using `anova` and the model with only the raw AllTrails search view counts had the lowest AIC.

```
##                                   Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## gamm_modGI_AR1_searchOnly_raw$lme     1 26  20412.5  20524.2  -10180.2
## gamm_modGI_AR1_Both_MA3$lme           2 27  13059.8  13175.9   -6502.9 1 vs 2   7354.6  <.0001
## gamm_modGI_AR1_StravaOnly$lme         3 26 761319.9 761431.6 -380633.9 2 vs 3 748262.0  <.0001
```

A look at deviance in these model predictions (Table A.1) shows a lot of variation between trails for the lowest amount of deviance.

Figure A.2 shows the variation in predictions when fitting models with different auxiliary data included.

A.4 Takeaway Conclusion

In our initial investigation of the AllTrails search views data we found evidence of a positive linear relationship with trail use (as camera counts) for only a few trails. We have shown evidence for including some form of AllTrails search views data in tandem with the Strava Metro provided daily trip counts in our model to improve predictive abilities. However, in our current approach, which fits a HGAMM to 21 different trail subsections, we would need AllTrails search data for all included trails. This model is unable to handle any missing covariate values. We do included AllTrails searches as the raw number of views in our Middle Cottonwood only approach (Section 5) but it was not a significant linear term.

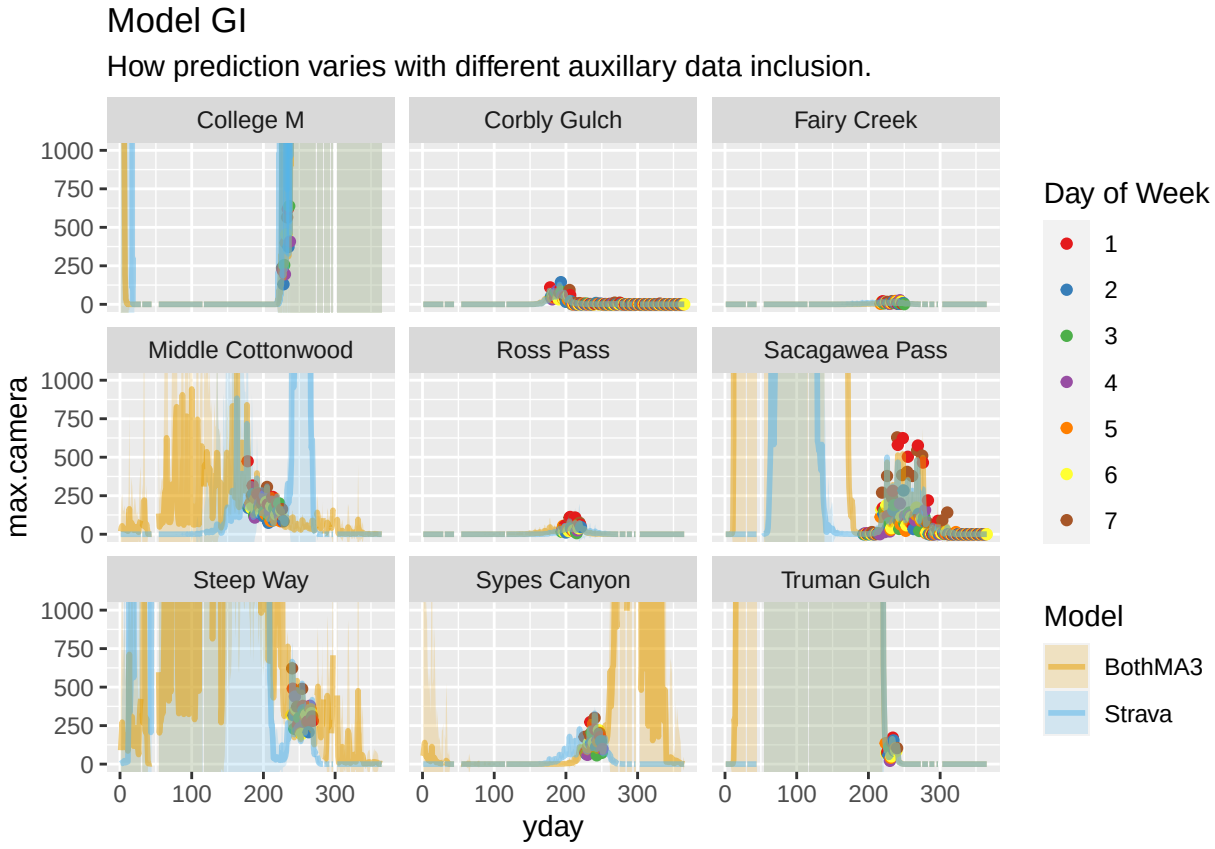


Figure A.2: Predicted trail use count values (lines) versus observed trail use (points) for each trail subsection, based on each model (Strava only with raw counts, both AllTrails and Strava with 3-day moving average, and Strava only).

Appendix B

Spatial Network Generalized Additive Mixture Model

```
## Reading layer 'bridger_trails_elev' from data source
##   '/Users/mlbartley/Documents/Consulting/Headwaters Economics/HE-TrailUse-R/data/raw/bridger_trails_e
##   using driver 'ESRI Shapefile'
## Simple feature collection with 47 features and 18 fields
## Geometry type: MULTILINESTRING
## Dimension:     XYM
## Bounding box:  xmin: -111.0979 ymin: 45.54056 xmax: -110.8828 ymax: 46.04659
## m_range:       mmin: -1.797693e+308 mmax: 20.16198
## Geodetic CRS:  NAD83
```

B.1 Overview of Potential Model

During our research into potential models for use in this statistical analysis we explored many options that ended up not being used in this report. One such model is presented here as it may prove more useful in future analyses.

We provided an overview of generalized additive models in Section 4 and in our analysis use a hierarchical generalized additive model with various temporal correlation structures to fit our trail use data. One drawback of this model is the difficulty of accounting for potential spatial correlation. The Bridger Mountains are home to a complex spatial network of trails some of which we have subdivided into subsections for this analysis. Several trails can be accessed by multiple trailheads, others are more traditional out-and-back or loops. It is a reasonable to assume that trail use would be more similar along subsections of a single trail and possibly between trails that are located closer together. Currently, we account for the random effect of subsection (`subsectionF`) in our models but we are unable to nest subsection within trail or include information about the overall spatial network.

Spatial network GAMs are another extension of the overall GAM framework that allows for the inclusion of a spatial network in a GAM approach that is used to create a spatial correlation structure in the model. Unfortunately, this is not currently an off-the-shelf way to also incorporate a **temporal** correlation structure. A bespoke spatiotemporal correlation structure could potentially created, but that is currently beyond the scope of this analysis. For the Bridger Mountain trail use statistical analysis it was determined that accounting for temporal correlation while including subsection as a random effect was the best way forward. We include an example of how to fit the spatial network model here in case future work may find it useful.

An overview of spatial network GAMs and two case studies with associated R code for analysis and plots are available at: https://github.com/nick-gauthier/gam-networks

B.2 Data used

The spatial data GAM requires the data be organized as a spatial network structure that includes (1) a network of "node" locations that mark the start and end locations of the different trail subsections and (2) the response and predictor variables with columns with location "edges" defined by "to" and "from" columns. For many spatial networks, these edges and nodes are easily defined. For example, towns as nodes and roads between them as edges. However, for a recreational trail network the edges are more clearly defined and the placement of nodes can require more thought. While the overall start and end nodes of an out-and-back trail might be easy to also define, other trails that branch and loop can prove more difficult. Spatial delineation of trails into subsections was decided by Headwaters Economics.

B.3 Model Fit

We fit this spatial GAM model using a correlation structure for symmetric relational data from the **corMLPE** package.

B.4 Model Diagnostics

The summary output for this `gamm` object shows an adjusted R-sq of 0.79 and a non-significant parametric term, `total_traveltime`.

A look at plots (see Figures **??** and **??**) that provide insight into the presence of temporal dependence remaining after the model is fit shows a high degree of temporal autocorrelation.

We have included additional diagnostic plots, however no additional issues are apparent beyond the temporal correlation.

B.5 Model Compare

This model includes all trail subsections, even ones with overlapping information about Bridger Rider, for example. So we are unable to directly compare this model with those used with only a single Bridger Ridge subsection as in models G, GS, and GI. However we determined that the amount of temporal autocorrelation was too high to use this model in our statistical analysis. Should a way to implement a combined spatiotemporal correlation strucutre is developed, we choose to move forward with a `gamm` approach with temporal autocorrelation structure and trail subsection as a random effect.

Bibliography

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., and Iannone, R. (2020). *rmarkdown: Dynamic Documents for R*. R package version 2.6.

Appelhans, T., Detsch, F., Reudenbach, C., and Woellauer, S. (2022). *mapview: Interactive Viewing of Spatial Data in R*. R package version 2.11.0.

Hyndman, R., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., and Yasmeen, F. (2022). *forecast: Forecasting functions for time series and linear models*. R package version 8.17.0.

Hyndman, R. J. and Khandakar, Y. (2008). Automatic time series forecasting: the forecast package for R. *Journal of Statistical Software*, 26(3):1–22.

Pedersen, E. J., Miller, D. L., Simpson, G. L., and Ross, N. (2019). Hierarchical generalized additive models in ecology: an introduction with mgcv. *PeerJ*, 7:e6876.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Simpson, G. L. (2022). *gratia: Graceful ggplot-Based Graphics and Other Functions for GAMs Fitted using mgcv*. R package version 0.7.3.

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., and Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43):1686.

Wickham, H. and Bryan, J. (2019). *readxl: Read Excel Files*. R package version 1.3.1.

Wickham, H., Chang, W., Henry, L., Pedersen, T. L., Takahashi, K., Wilke, C., Woo, K., Yutani, H., and Dunnington, D. (2020). *ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.3.3.

Wood, S. (2017). *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2 edition.

Wood, S., N., Pya, and S"afken, B. (2016). Smoothing parameter and model selection for general smooth models (with discussion). *Journal of the American Statistical Association*, 111:1548–1575.

Wood, S. N. (2003). Thin-plate regression splines. *Journal of the Royal Statistical Society (B)*, 65(1):95–114.

Wood, S. N. (2004). Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686.

Wood, S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semi-parametric generalized linear models. *Journal of the Royal Statistical Society (B)*, 73(1):3–36.

Wright, K. (2021). *pals: Color Palettes, Colormaps, and Tools to Evaluate Them*. R package version 1.7.

Xie, Y. (2020a). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.21.

Xie, Y. (2020b). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.30.

Xie, Y. (2022). *tinytex: Helper Functions to Install and Maintain TeX Live, and Compile LaTeX Documents*. R package version 0.41.

Zhu, H. (2021). *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*. R package version 1.3.4.