R-Ladies for PAWS Datathon (2019)

R-Ladies Philly

April 22, 2019

Executive Summary

- Most animals' outcomes are adoption, and the 'wait time' from intake to outcome is longer for cats than for dogs

- Younger age and poorer health contributes to longer wait time for cats; and intake and wait time for younger, health-compromised cats is highest in spring/summer
- Dogs showed no differences in wait time based on age, health, size, or season
- Increased resources in spring/summer months for young, unhealthy cats may shorten wait times and alleviate staff burden

Problem definition and dataset

The 2019 R-Ladies for PAWS Datathon aimed to help the Philadelphia Animal Welfare Society (PAWS) improve its adoptions processes. For this data challenge, PAWS made 2018 data available containing adoption application form submissions, staff processing of applications, and animal outcome data. We developed analytic approaches to better understand the following topics:

1. An animal's trajectory at PAWS
2. An adoption application's trajectory at PAWS
3. Geographic characteristics that influence adoptions
4. Social media activity that could influence adoptions

Timeline and Workflow

Prior to kickoff, data were obtained and preprocessed. After downloading the data from multiple sources, individual entries were matched based on first name, last name, address, and other relevant variables. Once matched, any identifiable information was removed from the dataset. This occurred before the data were made available to the group for analysis.

February 12: Kickoff Meetup: At this event, the project was introduced and teams were formed to work on one of the 4 topics outlined above. All participants were encouraged to join the R-Ladies Philly Slack workspace and to set up a GitHub account (which was used as the main collaborative platform).

After the kickoff meetup, groups worked together online, getting together on an as-needed basis. Questions were asked and answered via Slack, with an occasional clarification email to PAWS.

March 26: Conclusion Meetup: At this meetup, teams presented their results and discussed their experiences. Individual team reports were finalized in the weeks following this meetup, and then integrated into the present report.

Results

1. Animal Trajectories

This analysis investigated factors relating to an animal's trajectory in the PAWS system using PetPoint data from 2018. The group operationalized animal trajectory as wait time and outcome (e.g. adoption), with wait time defined as time in days from intake to outcome. We restricted analyses to dogs and cats, as other animals' data points were sparse and compromised statistical power. Our primary factors of interest included animal characteristics (size, breed, health), intake type, and seasonal patterns.

Contributors

Alex Lesicko, PhD is a postdoctoral fellow studying auditory coding at the University of Pennsylvania. She recently moved to Philadelphia from Chicago, where she completed her PhD in neuroscience.

Jake Riley is a clinical data analyst at the Children's Hospital of Philadelphia (CHOP). He enjoys developing tools for analytic teams and specializes on data visualization and geospatial information systems (GIS).

Javier Jasso is a certified speech–language pathologist and a PhD candidate in communication sciences and disorders at the University of Texas at Austin. Javier has expertise in the assessment of culturally/linguistically diverse children, focusing on bilingual language acquisition.

Katerina Placek is a PhD candidate in neuroscience at the University of Pennsylvania and a co-organizer of R-Ladies Philly. She enjoys integrating outreach with teaching and learning in the local data science community.

Results

Our analyses focused on four facets of the PetPoint dataset:
1. Data Exploration and defining 'Wait Time'
2. Animal Characteristics
3. Intake and Outcome Characteristics
4. Seasonal/Locational Patterns

1. Data Exploration and Defining 'Wait Time'

We first examined animals' outcomes for all animals in the 2018 Pet Point datset where information was available and the outcome was not "Euthanasia" or "Died". Of the 2831 animals, 2777 were adopted, 30 were returned to their owner or guardian, and 24 were transferred out. With regard to animal species, there were 2424 cats (2390 adopted, 18 returned to owner, and 16 transferred out) and 407 dogs (387 adopted, 12 returned to owner, and 8 transferred out). Overall, more than 95% of animals who had a live outcome were placed with an adoptive family via PAWS.

The median 'wait time' (time in days from intake to outcome) for an animal at PAWS in 2018 was 45 days (51 days for cats, and 18 days for dogs). More specifically, cats waited approx. 18.5 days to be transferred out, 4.5 days to be reunited with their owner, and 52 days to find an adopter, while dogs waited 1.5 days to be transferred out, 2 days to be reunited with their owner, and 20 days to be adopted.

Next, for each species, we visualized which PetPoint variables contributed to differences in wait time, and focused our subsequent analyses on the variables with the greatest contributions to wait times (i.e. the longest horizontal lines in the two plots below):

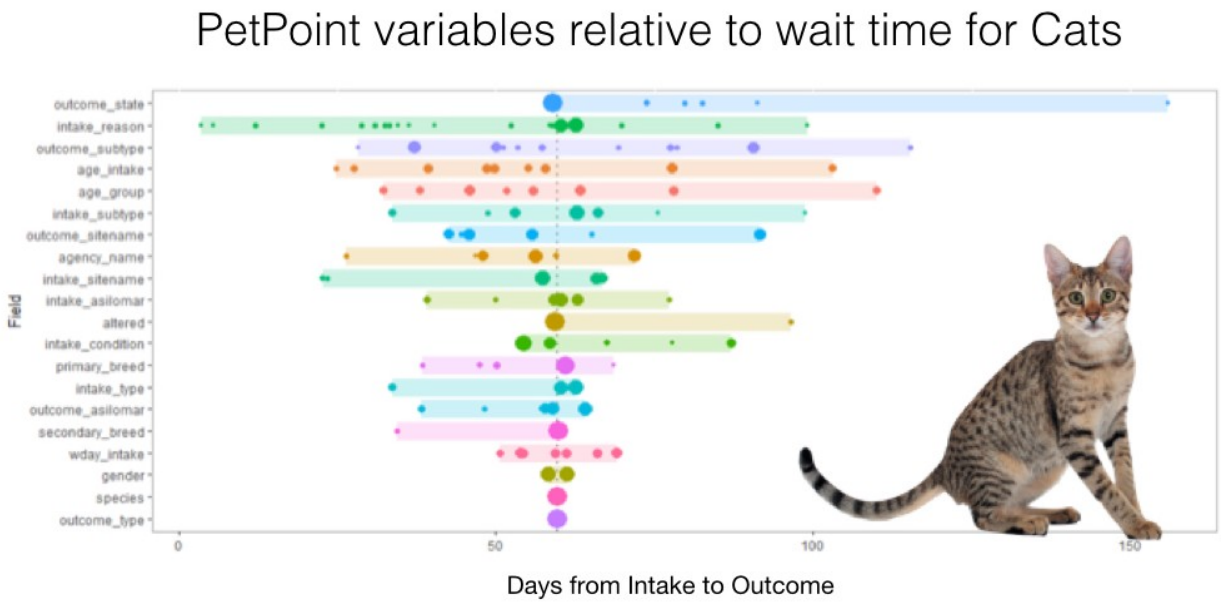


Figure 1:

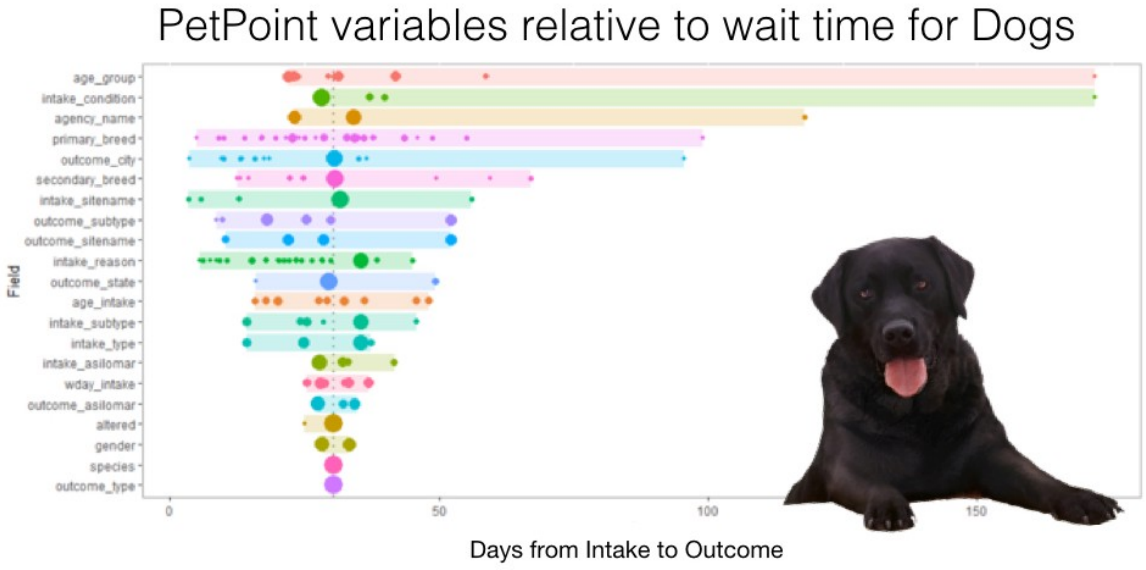


Figure 2:

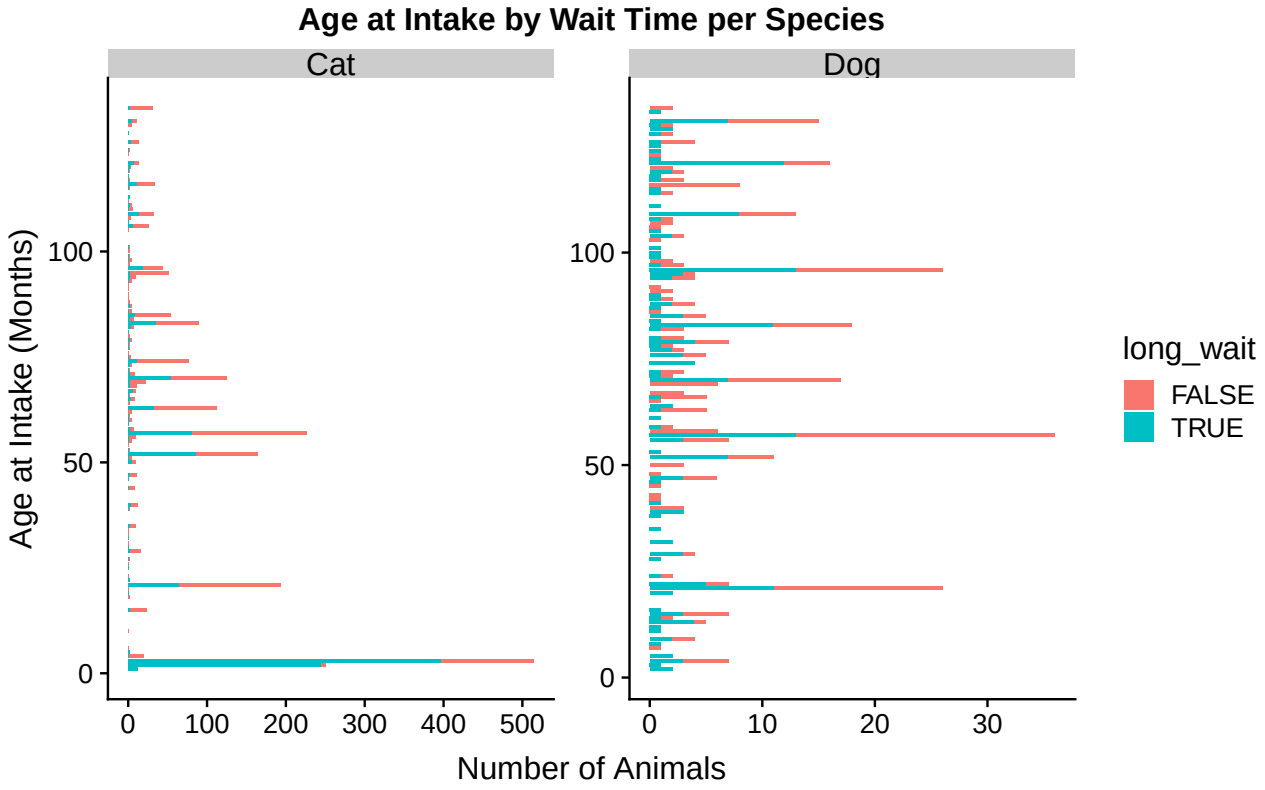
2. Animal Characteristics

For animal characteristics contributing to wait time at PAWS, we first examined breed per species. We found that most, if not all, cats from PAWS were 'domestic short hair', while dogs' breed variability was greater. Among dogs, our findings indicate that Shih Tzus tend to have shorter wait times whereas Terriers tend to have longer wait times.

Given the large number of unique dog breeds in the PetPoint dataset, it was difficult to draw definitive conclusions from the above result. Therefore, we classified dogs into 3 size categories based on average weight per breed. Our analyses demonstrated no significant differences in wait time based on dog size category.

We then examined age group per species, to determine whether wait time different on age group for dogs and cats:

```
# plot n per age group
raw_data %>%
  ggplot(aes(x = age_intake, fill = long_wait)) +
  geom_bar() +
  facet_wrap(~species, scales = "free") +
  coord_flip()+
  xlab("Age at Intake (Months)") +
  ylab("Number of Animals") +
theme(strip.text = element_text(size=14))+
        ggtitle("Age at Intake by Wait Time per Species")
```



We found that cats in younger age groups have longer wait times, whereas but found no differences in dogs based on age group.

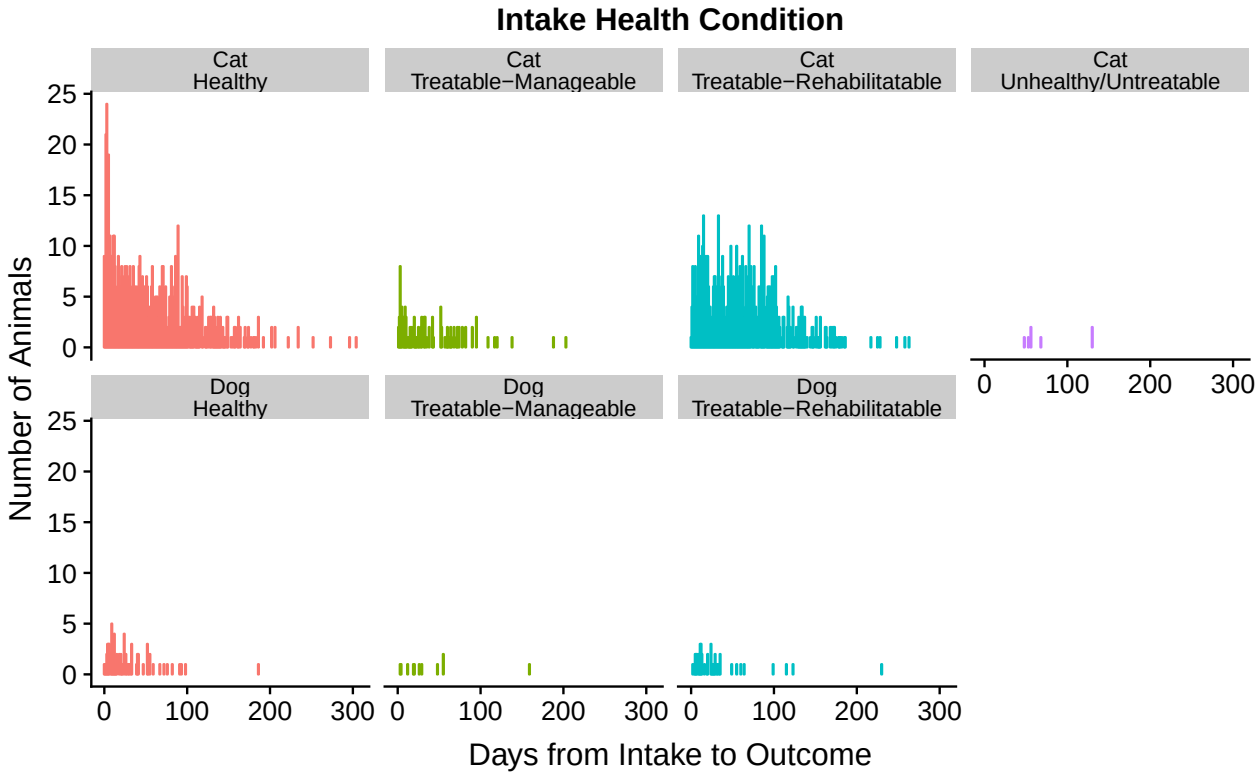
Last, we examined health at intake and outcome per species relative to wait times.

```
  raw_data %>%
    filter(!is.na(intake_asilomar))%>%
```

```
    group_by(intake_asilomar, species) %>%
    mutate(median = median(wait_days, na.rm =)) %>%
    group_by(Species = species, "Intake Asilomar" = intake_asilomar, "Median Wait Time (Days)" = median
    summarise("Number of Animals"=n()) %>%
    kable()
```

Species	Intake Asilomar	Median Wait Time (Days)	Number of Animals
Cat	Healthy	47	694
Cat	Treatable-Manageable	29	104
Cat	Treatable-Rehabilitatable	57	652
Cat	Unhealthy/Untreatable	56	7
Dog	Healthy	20	73
Dog	Treatable-Manageable	26	11
Dog	Treatable-Rehabilitatable	20	46

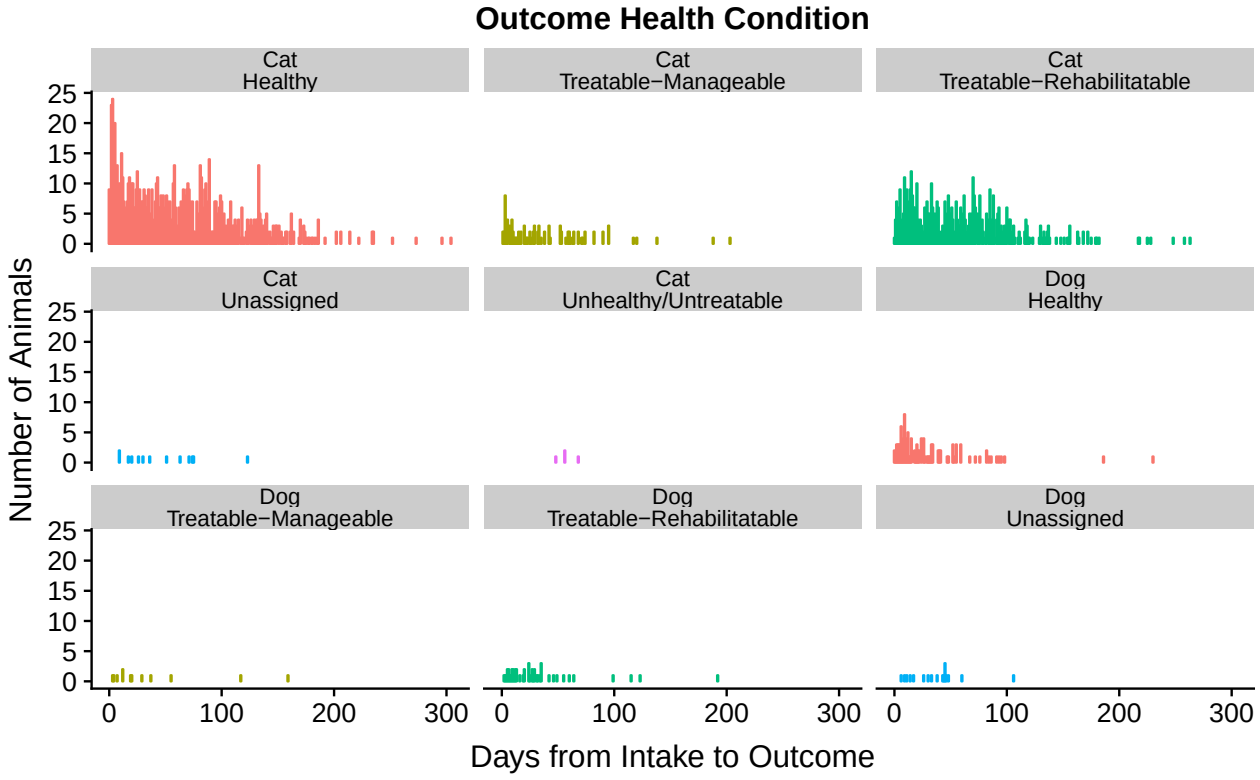
```
# plot n per intake health condition
  raw_data %>%
    filter(!is.na(intake_asilomar))%>%
    group_by(intake_asilomar, species) %>%
    group_by(species, wait_days, intake_asilomar) %>%
    summarise(n=n()) %>%
    ggplot(aes(x = wait_days, y = n, col = intake_asilomar)) +
        geom_col() +
        xlab("Days from Intake to Outcome") +
        ylab("Number of Animals") +
        facet_wrap(species~intake_asilomar, scales = "fixed", ncol = 4)+
        theme(strip.text = element_text(size=10), legend.position = "none") +
        ggtitle("Intake Health Condition")
```



```r
raw_data %>%
    filter(!is.na(intake_asilomar))%>%
    group_by(intake_asilomar, species) %>%
    mutate(median = median(wait_days, na.rm =)) %>%
    group_by(Species = species, "Outcome Asilomar" = intake_asilomar, "Median Wait Time (Days)" = median
    summarise("Number of Animals"=n()) %>%
    kable()
```

Species	Outcome Asilomar	Median Wait Time (Days)	Number of Animals
Cat	Healthy	47	694
Cat	Treatable-Manageable	29	104
Cat	Treatable-Rehabilitatable	57	652
Cat	Unhealthy/Untreatable	56	7
Dog	Healthy	20	73
Dog	Treatable-Manageable	26	11
Dog	Treatable-Rehabilitatable	20	46

```r
# plot n per outcome health condition
raw_data %>%
    filter(!is.na(outcome_asilomar))%>%
    group_by(species, wait_days, outcome_asilomar) %>%
    summarise(n=n()) %>%
    ggplot(aes(x = wait_days, y = n, col = outcome_asilomar)) +
        geom_col() +
        xlab("Days from Intake to Outcome") +
        ylab("Number of Animals") +
        facet_wrap(species~outcome_asilomar, scales = "fixed") +
        theme(strip.text = element_text(size=10), legend.position = "none")+
        ggtitle("Outcome Health Condition")
```



Our analyses revealed that for cats only, health condition at intake and outcome was associated with longer median wait time. Specifically, cats classified as 'Treatable-Rehabilitatable' and 'Unhealthy/Untreatable' had longer wait times than cats classified as 'Healthy' or 'Treatable-Manageable'.

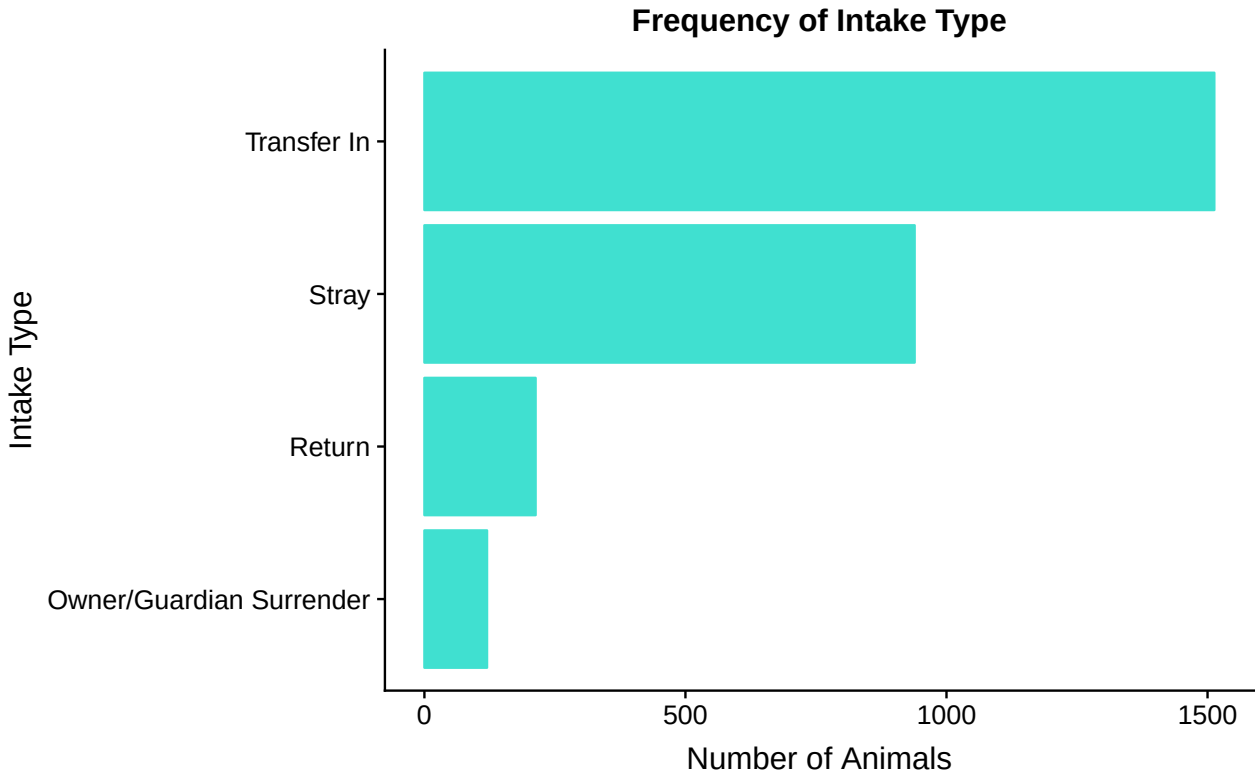
3. Intake Characteristics

Next, we examined intake characteristics across the petpoint dataset.

```
master_animal_intake <- master_animal %>%
    dplyr::select(intake_type,
           intake_subtype,
           wait_days,
           animal_type) %>%
    filter(!is.na(intake_type)) %>%
    mutate(intake_type = factor(intake_type),
           animal_type = factor(animal_type),
           wait = as.numeric(wait_days))
```

We first examined the frequency of primary intake type for all animals and calculated the median wait days for all animals based on intake type:

```
ggplot(master_animal_intake, aes(intake_type))+
    geom_bar(color="turquoise", fill="turquoise")+
    xlab('Intake Type') +
    ylab('Number of Animals')+
    coord_flip() +
    ggtitle("Frequency of Intake Type")
```



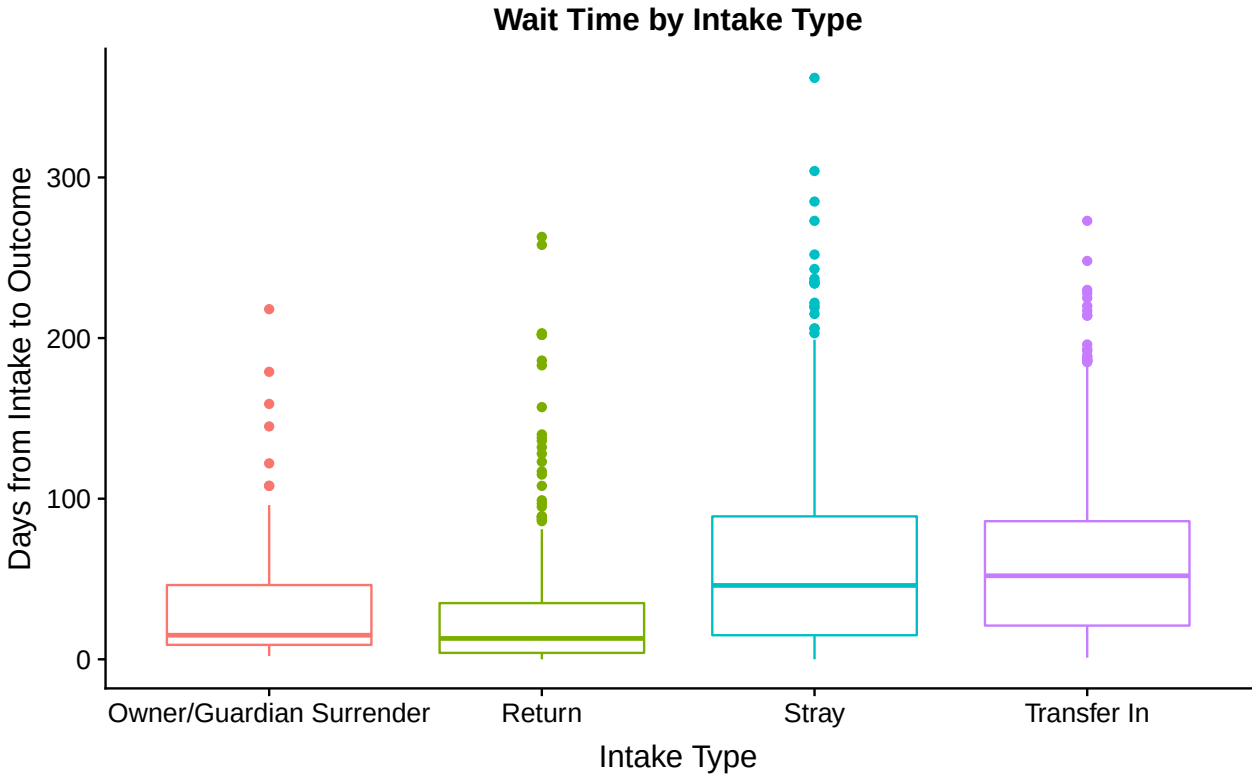

```
master_animal_intake %>%
    group_by("Intake Type" = intake_type) %>%
```

```
    summarise("Number of Animals" = n(), "Median Wait Time (Days)" = median(wait_days)) %>%
    kable()
```

Intake Type	Number of Animals	Median Wait Time (Days)
Owner/Guardian Surrender	120	15
Return	213	13
Stray	939	46
Transfer In	1513	52

Using analysis of variance, we examined whether wait days differed based on animal intake type:

```
intake_aov <- aov(wait_days ~ intake_type, master_animal_intake)
#summary(intake_aov)

ggplot(master_animal_intake, aes(x = intake_type, y = wait_days, col = intake_type)) +
    geom_boxplot() +
    xlab('Intake Type') +
    ylab('Days from Intake to Outcome')+
    theme(legend.position = "none") +
    ggtitle("Wait Time by Intake Type")
```



Our results indicated that animals at PAWS with an intake of "Stray" or "Transfer In" had significantly longer wait times relative to animals who with an intake type of "Owner/Guardian Surrender" or "Return."

We next examined intake type by species:

```
master_animal_intake %>%
    group_by("Intake Type" = intake_type) %>%
    filter(animal_type=='cat') %>%
    summarise("Number of Cats" = n(), "Median Wait Time (Days)" = median(wait_days))%>%
    kable()
```

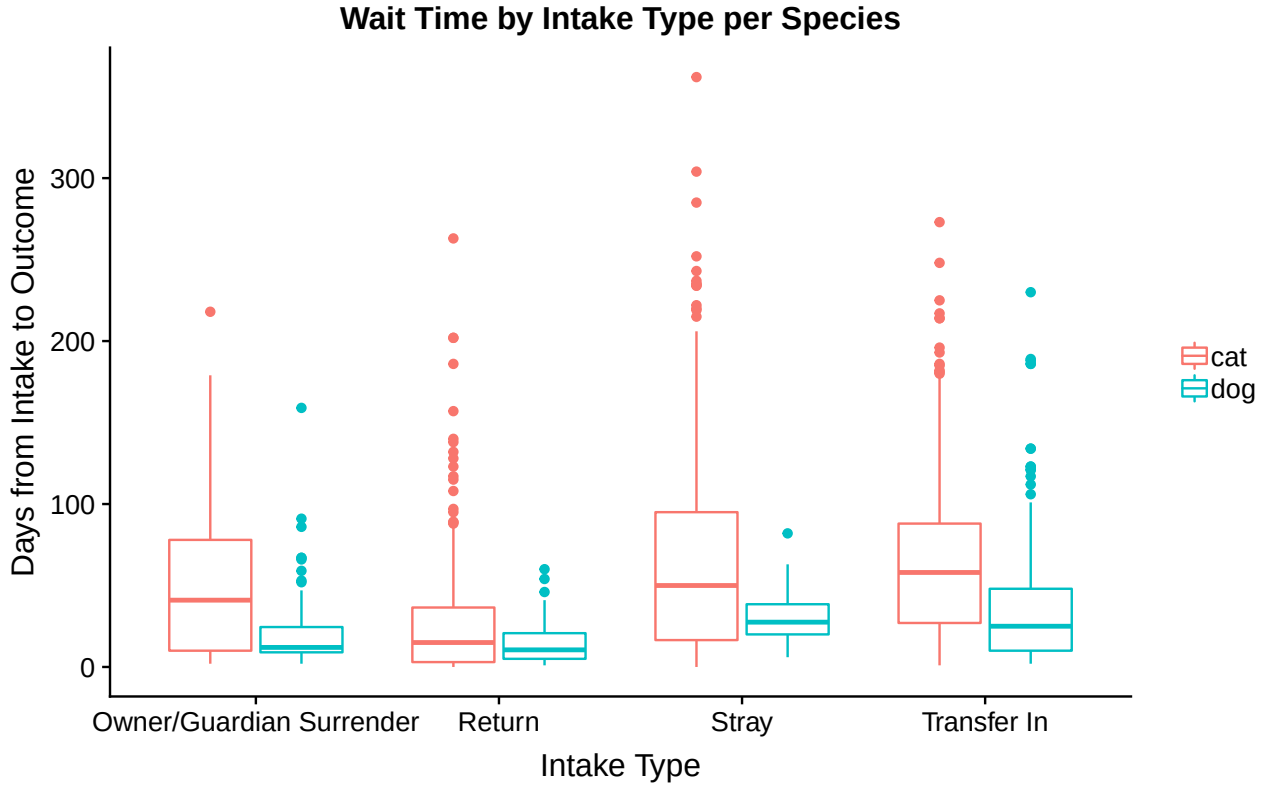
Intake Type	Number of Cats	Median Wait Time (Days)
Owner/Guardian Surrender	33	41
Return	134	15
Stray	771	50
Transfer In	1068	58

```
master_animal_intake %>%
     group_by("Intake Type" = intake_type) %>%
    filter(animal_type == 'dog') %>%
     summarise("Number of Dogs" = n(), "Median Wait Time (Days)"  = median(wait_days))%>%
    kable()
```

Intake Type	Number of Dogs	Median Wait Time (Days)
Owner/Guardian Surrender	64	12.0
Return	42	10.5
Stray	18	27.5
Transfer In	197	25.0

And we examined whether the effect of intake type on wait time differed by animal species:

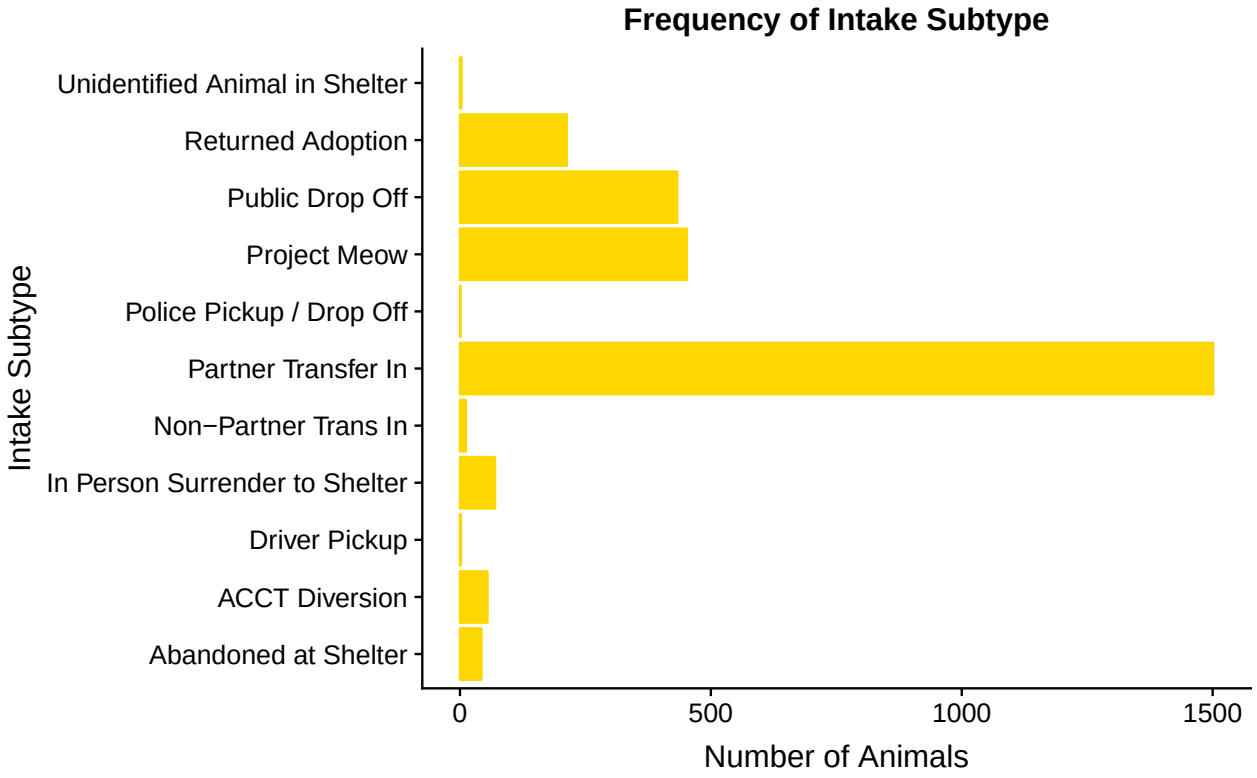
```
intake_type_int <- lm(wait_days ~ intake_type*animal_type, master_animal_intake)
#summary(intake_type_int)

master_animal_intake %>%
filter(!is.na(animal_type)) %>%
ggplot(., aes(x = intake_type, y = wait_days, col = animal_type))+
    geom_boxplot() +
    xlab('Intake Type') +
    ylab('Days from Intake to Outcome')+
    ggtitle("Wait Time by Intake Type per Species") +
    theme(legend.title = element_blank())
```



We found that as previously, cats had longer wait times for each intake type relative to dogs.

We next examined the intake subtype for all animals:

```
#plotting frequency of intake_subtype var:
ggplot(master_animal_intake, aes(intake_subtype)) +
    geom_bar(color="gold", fill="gold")+
    xlab('Intake Subtype') +
    ylab('Number of Animals')+
    coord_flip()+
    ggtitle("Frequency of Intake Subtype")
```



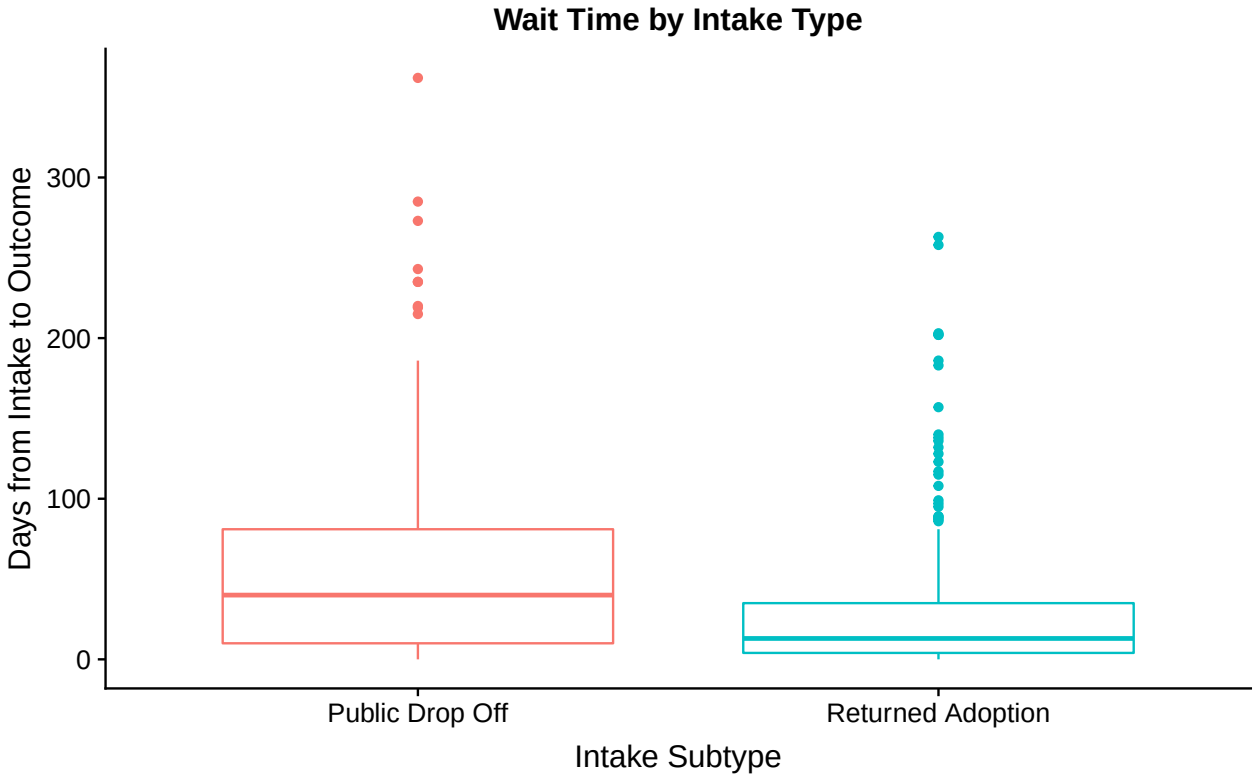

```
master_animal_intake %>%
    group_by("Intake Subtype" = intake_subtype) %>%
    summarise("Number of Animals" = n(), "Median Wait Time (Days)" = median(wait_days)) %>%
    kable()
```

Intake Subtype	Number of Animals	Median Wait Time (Days)
Abandoned at Shelter	43	30.0
ACCT Diversion	55	15.0
Driver Pickup	2	39.0
In Person Surrender to Shelter	70	17.5
Non-Partner Trans In	12	62.5
Partner Transfer In	1501	52.0
Police Pickup / Drop Off	1	57.0
Project Meow	452	56.5
Public Drop Off	433	40.0
Returned Adoption	213	13.0
Unidentified Animal in Shelter	3	58.0

We focused our analysis of intake subtype on animals in the Public Dropoff and Returned Adoption categories:

```r
subtype_filter <- master_animal_intake %>%
    filter(intake_subtype == 'Returned Adoption' |
               intake_subtype == 'Public Drop Off')
```

```r
intake_subtype_lm <- lm(wait_days ~ intake_subtype, subtype_filter)
#summary(intake_subtype_lm)

ggplot(subtype_filter, aes(x = intake_subtype, y = wait_days, col = intake_subtype)) +
    geom_boxplot() +
    xlab('Intake Subtype') +
    ylab('Days from Intake to Outcome')+
    ggtitle("Wait Time by Intake Type") +
    theme(legend.position = "none")
```



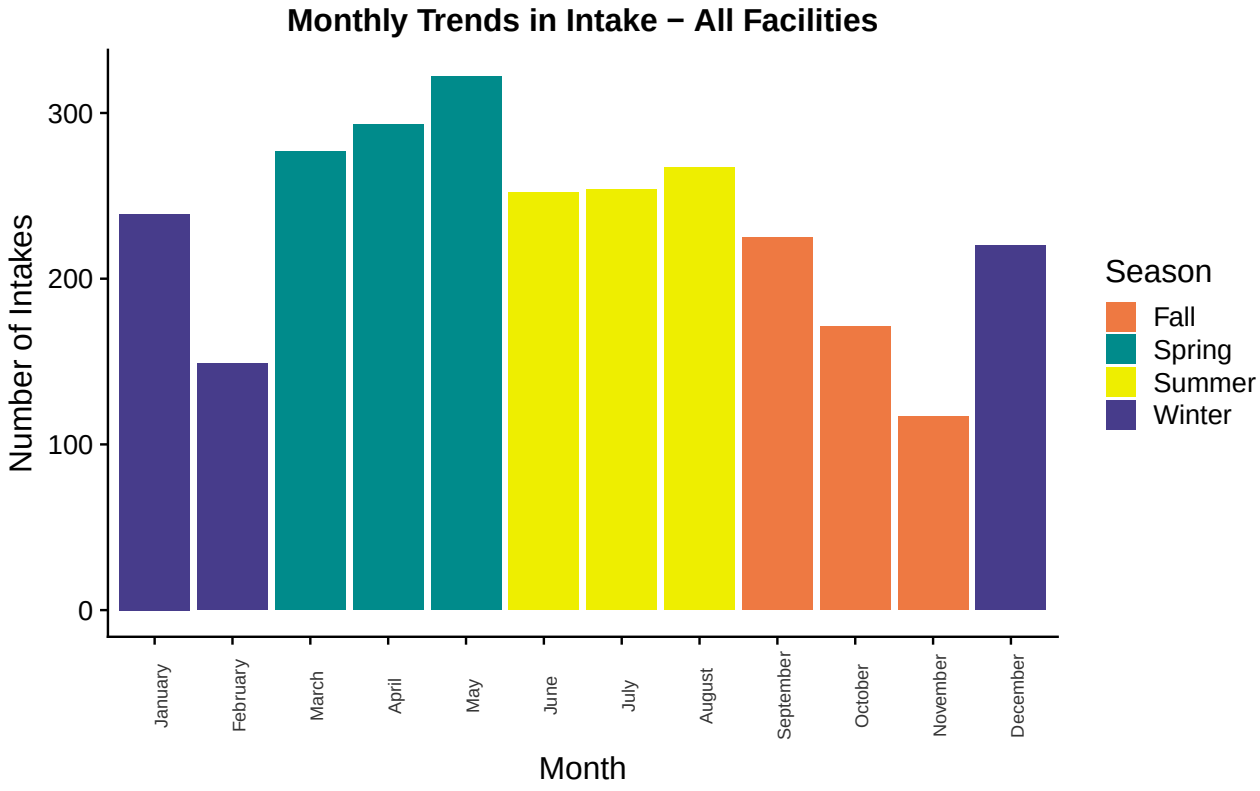
We found that animals with an intake subtype of 'Public Drop Off' had a significantly longer wait time than animals with an intake subtype of 'Returned Adoption.'

4. Seasonal and Locational Patterns

Last, we examined animals' wait time at PAWS by season and location. We examined the frequency of intakes per month:

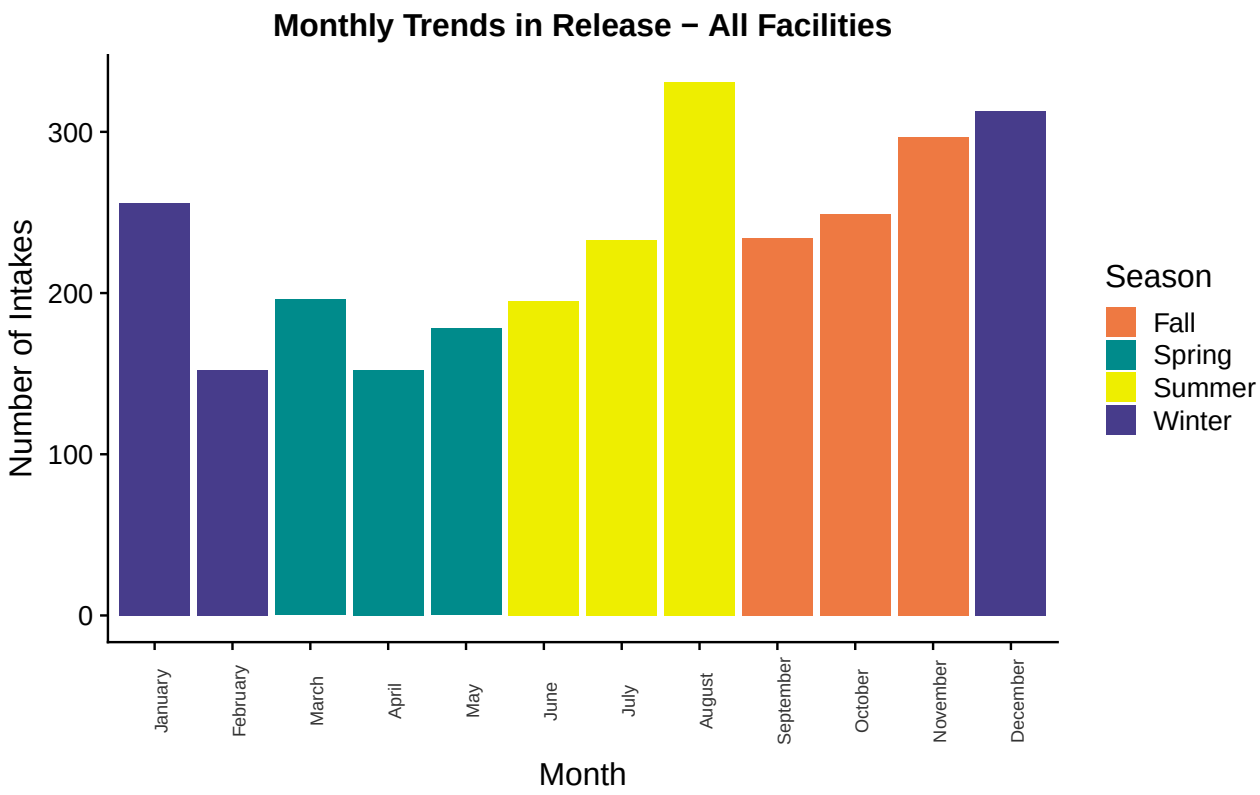
```r
#Extract intake month
master_animal$intake_month <- format(as.Date(master_animal$intake_date), "%B")
#Plot monthly trends in intake
master_animal$intake_month_fac = factor(master_animal$intake_month, levels = month.name)
intake_month_counts <- table(master_animal$intake_month_fac)
df_intake_month_counts <- data.frame(intake_month_counts)
names(df_intake_month_counts)[1]<-"Month"
```

```
df_intake_month_counts$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("Fall",3), "W

ggplot(df_intake_month_counts, aes(x=Month, y=Freq, fill = Season)) +
    geom_bar(stat = "identity") +
    scale_fill_manual(values = c("Winter" = "slateblue4", "Spring" = "cyan4", "Summer" = "yellow2", "Fa
    ggtitle("Monthly Trends in Intake - All Facilities") + theme(plot.title = element_text(hjust = 0.5)
    ylab("Number of Intakes")
```



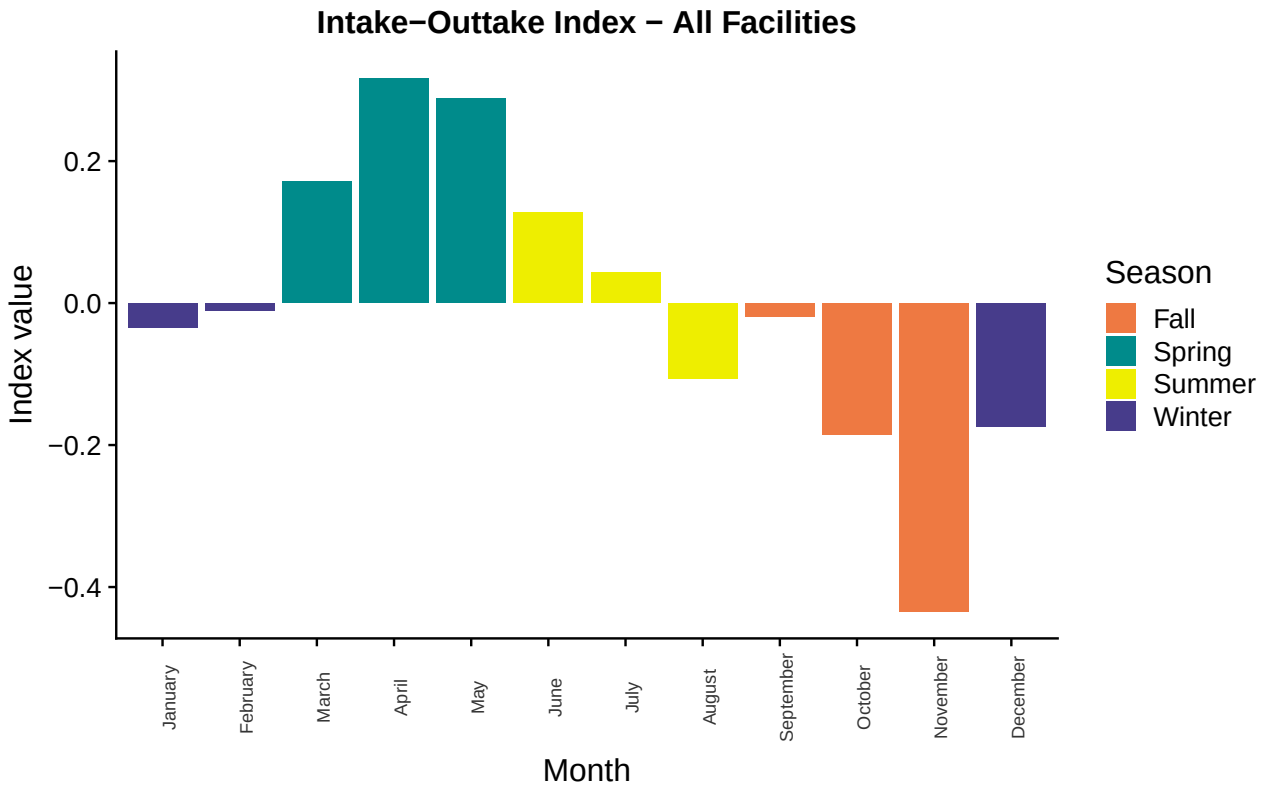
And examined the frequency of releases per month:

```
#Extract release month
master_animal$release_month <- format(as.Date(master_animal$release_date), "%B")

#Plot monthly trends in release
master_animal$release_month_fac = factor(master_animal$release_month, levels = month.name)
release_month_counts <- table(master_animal$release_month_fac)
df_release_month_counts <- data.frame(release_month_counts)
names(df_release_month_counts)[1]<-"Month"
df_release_month_counts$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("Fall",3), "

ggplot(df_release_month_counts, aes(x=Month, y=Freq, fill = Season)) +
    geom_bar(stat = "identity") +
    scale_fill_manual(values = c("Winter" = "slateblue4", "Spring" = "cyan4", "Summer" = "yellow2", "Fa
    ggtitle("Monthly Trends in Release - All Facilities") + theme(plot.title = element_text(hjust = 0.5
    ylab("Number of Intakes")
```



And calculated the difference between intakes and releases per month:

```
#Plot intake-outtake index
df_intake_outtake_index <- cbind(df_intake_month_counts[1],df_intake_month_counts[3])

df_intake_outtake_index$Index <-(df_intake_month_counts$Freq-df_release_month_counts$Freq)/(df_intake_mo

ggplot(df_intake_outtake_index, aes(x=Month, y=Index, fill = Season)) +
    geom_bar(stat = "identity") +
    scale_fill_manual(values = c("Winter" = "slateblue4", "Spring" = "cyan4", "Summer" = "yellow2", "Fal
    ggtitle("Intake-Outtake Index - All Facilities") + theme(plot.title = element_text(hjust = 0.5), axi
    ylab("Index value")
```

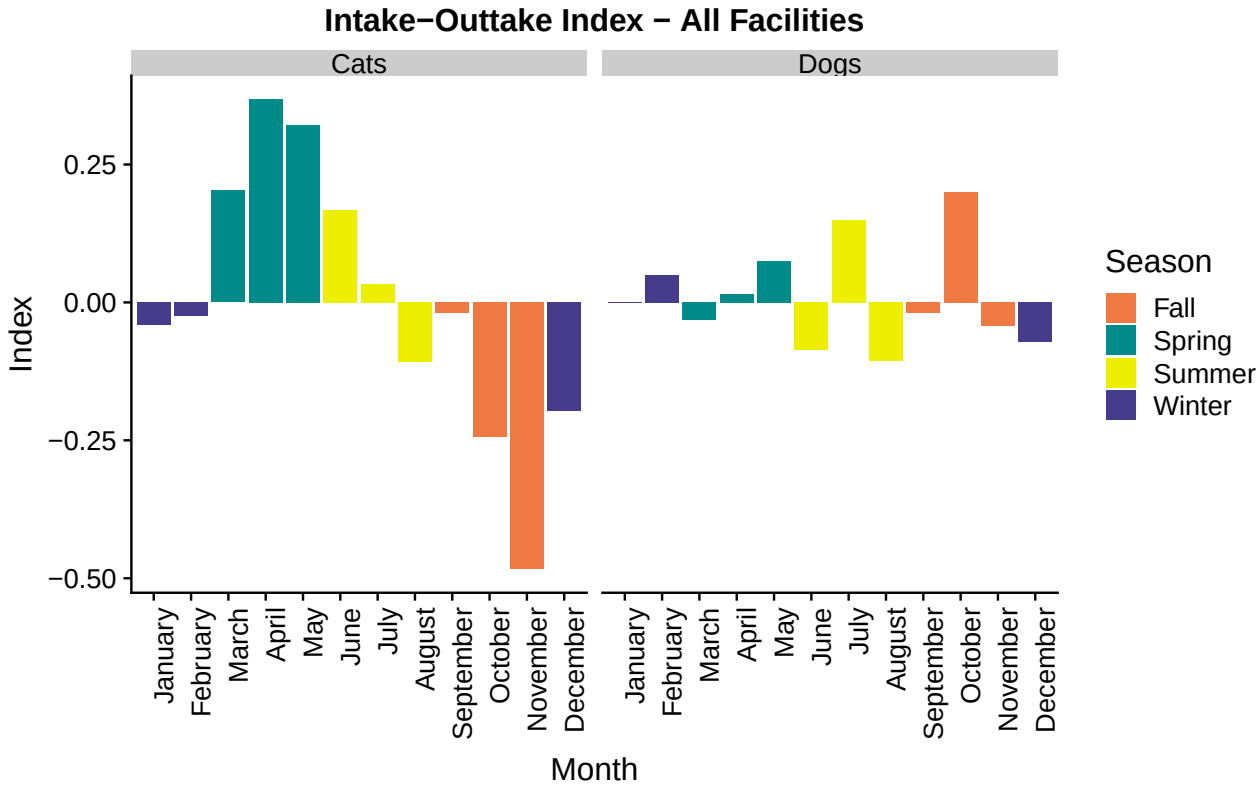


We found that the summer and spring months have higher intakes relative to releases.

We then looked at monthly trends in intakes and releases by species:

```
#Subset Data by Species
cat <- master_animal[ which(master_animal$species=='Cat'), ]
dog <- master_animal[ which(master_animal$species=='Dog'), ]

#calculate monthly trends in intake by species
cat_intake_month_counts <- table(cat$intake_month_fac)
dog_intake_month_counts <- table(dog$intake_month_fac)
df_intake_month_counts_species <- data.frame(cat_intake_month_counts, dog_intake_month_counts)
df_intake_month_counts_species <- df_intake_month_counts_species[-3]
names(df_intake_month_counts_species)[1] <- "Month"
names(df_intake_month_counts_species)[2] <- "Cats"
names(df_intake_month_counts_species)[3] <- "Dogs"

df_intake_month_counts_species <- melt(df_intake_month_counts_species, id.vars='Month')
df_intake_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("Fall

#calculate monthly trends in release by species
cat_release_month_counts <- table(cat$release_month_fac)
dog_release_month_counts <- table(dog$release_month_fac)
df_release_month_counts_species <- data.frame(cat_release_month_counts, dog_release_month_counts)
df_release_month_counts_species <- df_release_month_counts_species[-3]
names(df_release_month_counts_species)[1] <- "Month"
names(df_release_month_counts_species)[2] <- "Cats"
names(df_release_month_counts_species)[3] <- "Dogs"

df_release_month_counts_species <- melt(df_release_month_counts_species, id.vars='Month')
```

```r
df_release_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("Fall

#calculate monthly trends in intake-outtake index by species
df_intake_outtake_index_species <- cbind(df_intake_month_counts_species[1],df_intake_month_counts_speci

df_intake_outtake_index_species$Index <-(df_intake_month_counts_species$value-df_release_month_counts_sp

ggplot(df_intake_outtake_index_species, aes(x=Month, y=Index, fill = Season)) + geom_bar(stat='identity
```



We found that cats have higher intakes than releases in spring and summer months, but observed no seasonal patterns for dogs.

We then examined seasonal patterns in intake-release index by location for each species:

```r
#Subset Data by Intake Location
GFA <- master_animal[ which(master_animal$intake_sitename=='Grays Ferry Avenue'), ]
PAC <- master_animal[ which(master_animal$intake_sitename=='PAC'), ]
GA <- master_animal[ which(master_animal$intake_sitename=='Grant Avenue'), ]
PAWS_FP <- master_animal[ which(master_animal$intake_sitename=='PAWS Foster Program'), ]
PAWS_OA <- master_animal[ which(master_animal$intake_sitename=='PAWS Offsite Adoptions'), ]
```

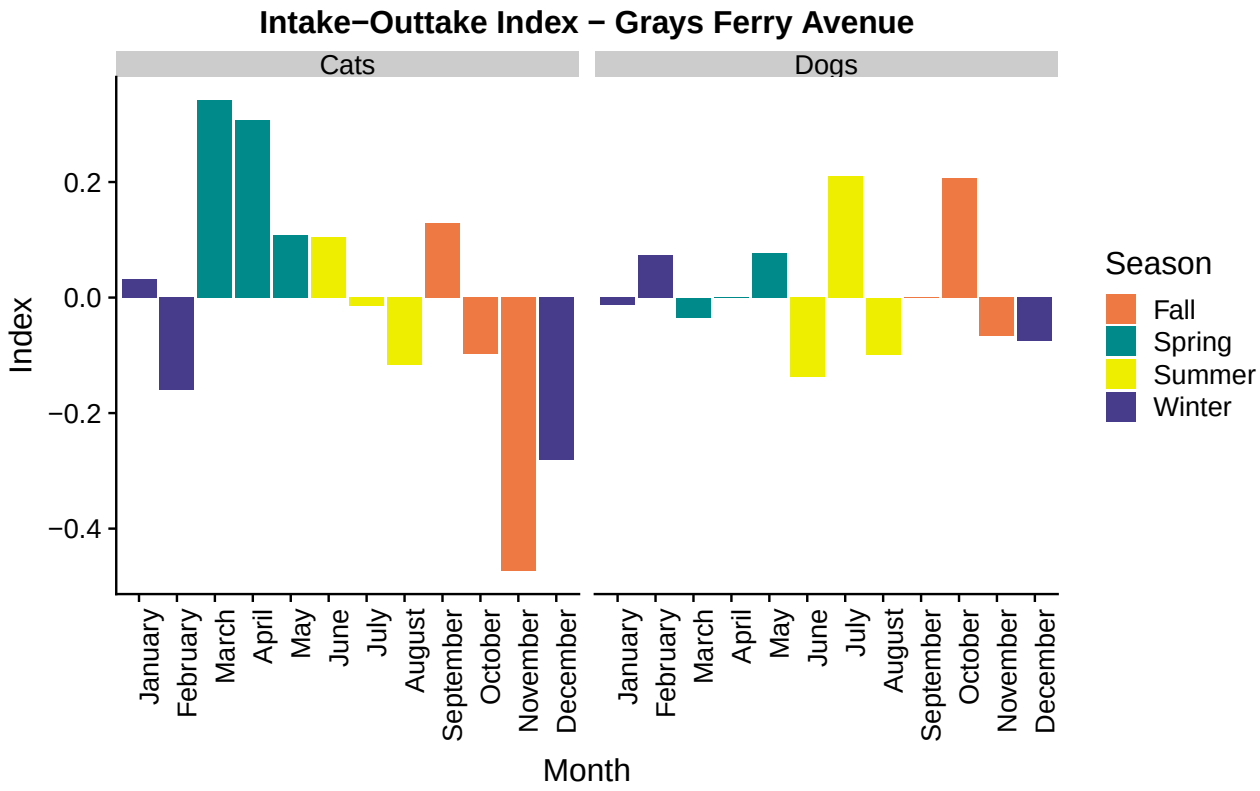
```r
#Subset Intake Location Data by Species
GFA_cat <- GFA[ which(GFA$species=='Cat'), ]
GFA_dog <- GFA[ which(GFA$species=='Dog'), ]

PAC_cat <- PAC[ which(PAC$species=='Cat'), ]
PAC_dog <- PAC[ which(PAC$species=='Dog'), ]

GA_cat <- GA[ which(GA$species=='Cat'), ]
GA_dog <- GA[ which(GA$species=='Dog'), ]
```

```
PAWS_FP_cat <- PAWS_FP[ which(PAWS_FP$species=='Cat'), ]
PAWS_FP_dog <- PAWS_FP[ which(PAWS_FP$species=='Dog'), ]

PAWS_OA_cat <- PAWS_OA[ which(PAWS_OA$species=='Cat'), ]
PAWS_OA_dog <- PAWS_OA[ which(PAWS_OA$species=='Dog'), ]
```

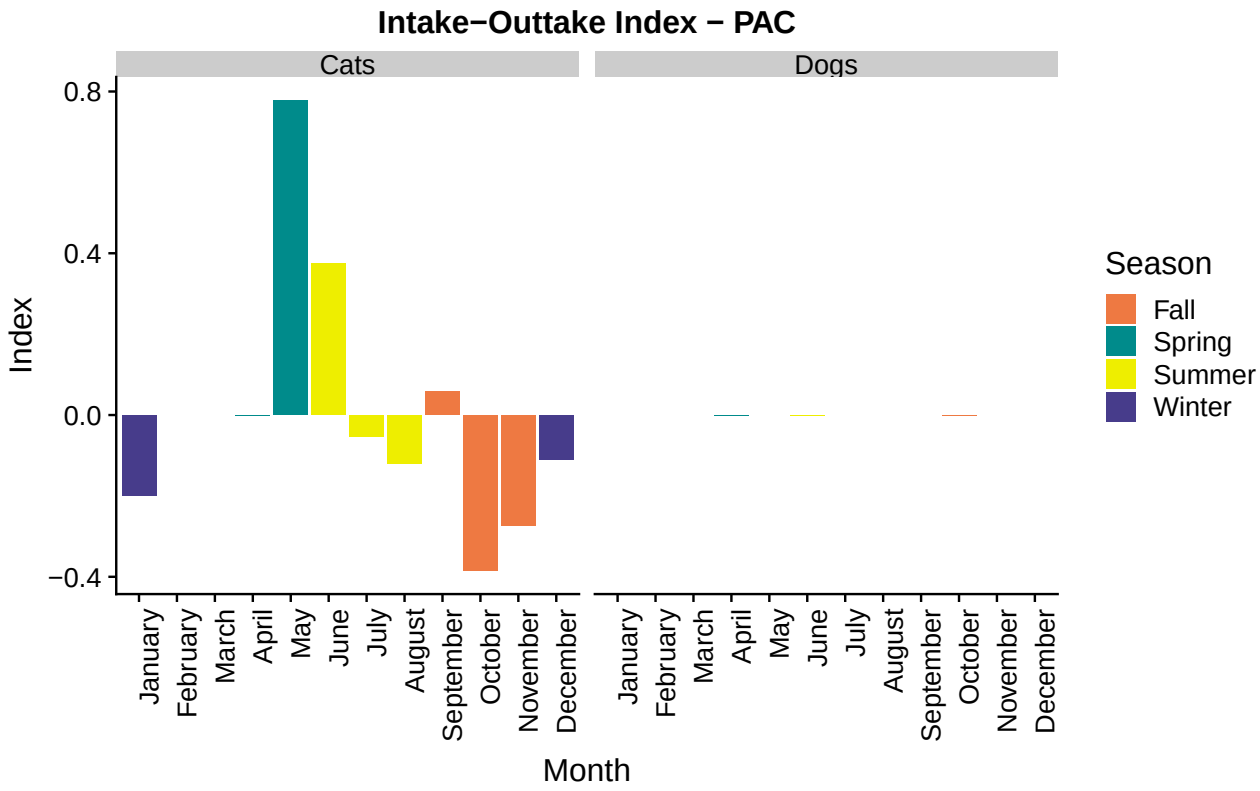
```
#Plot Monthly Trends in Intake-Release Index by Species for Each Location

#Grays Ferry Avenue
GFA_cat_intake_month_counts <- table(GFA_cat$intake_month_fac)
GFA_dog_intake_month_counts <- table(GFA_dog$intake_month_fac)
df_GFA_intake_month_counts_species <- data.frame(GFA_cat_intake_month_counts, GFA_dog_intake_month_coun
df_GFA_intake_month_counts_species <- df_GFA_intake_month_counts_species[-3]
names(df_GFA_intake_month_counts_species)[1] <- "Month"
names(df_GFA_intake_month_counts_species)[2] <- "Cats"
names(df_GFA_intake_month_counts_species)[3] <- "Dogs"

df_GFA_intake_month_counts_species <- melt(df_GFA_intake_month_counts_species, id.vars='Month')
df_GFA_intake_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("

#PAC
PAC_cat_intake_month_counts <- table(PAC_cat$intake_month_fac)
PAC_dog_intake_month_counts <- table(PAC_dog$intake_month_fac)
df_PAC_intake_month_counts_species <- data.frame(PAC_cat_intake_month_counts, PAC_dog_intake_month_coun
df_PAC_intake_month_counts_species <- df_PAC_intake_month_counts_species[-3]
names(df_PAC_intake_month_counts_species)[1] <- "Month"
names(df_PAC_intake_month_counts_species)[2] <- "Cats"
names(df_PAC_intake_month_counts_species)[3] <- "Dogs"

df_PAC_intake_month_counts_species <- melt(df_PAC_intake_month_counts_species, id.vars='Month')
df_PAC_intake_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("

#Grant Avenue
GA_cat_intake_month_counts <- table(GA_cat$intake_month_fac)
GA_dog_intake_month_counts <- table(GA_dog$intake_month_fac)
df_GA_intake_month_counts_species <- data.frame(GA_cat_intake_month_counts, GA_dog_intake_month_counts)
df_GA_intake_month_counts_species <- df_GA_intake_month_counts_species[-3]
names(df_GA_intake_month_counts_species)[1] <- "Month"
names(df_GA_intake_month_counts_species)[2] <- "Cats"
names(df_GA_intake_month_counts_species)[3] <- "Dogs"

df_GA_intake_month_counts_species <- melt(df_GA_intake_month_counts_species, id.vars='Month')
df_GA_intake_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("Fa

#PAWS Foster Program
PAWS_FP_cat_intake_month_counts <- table(PAWS_FP_cat$intake_month_fac)
PAWS_FP_dog_intake_month_counts <- table(PAWS_FP_dog$intake_month_fac)
df_PAWS_FP_intake_month_counts_species <- data.frame(PAWS_FP_cat_intake_month_counts, PAWS_FP_dog_intake
df_PAWS_FP_intake_month_counts_species <- df_PAWS_FP_intake_month_counts_species[-3]
names(df_PAWS_FP_intake_month_counts_species)[1] <- "Month"
names(df_PAWS_FP_intake_month_counts_species)[2] <- "Cats"
names(df_PAWS_FP_intake_month_counts_species)[3] <- "Dogs"
```

```r
df_PAWS_FP_intake_month_counts_species <- melt(df_PAWS_FP_intake_month_counts_species, id.vars='Month')
df_PAWS_FP_intake_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), re


#PAWS Offsite Adoptions
PAWS_OA_cat_intake_month_counts <- table(PAWS_OA_cat$intake_month_fac)
PAWS_OA_dog_intake_month_counts <- table(PAWS_OA_dog$intake_month_fac)
df_PAWS_OA_intake_month_counts_species <- data.frame(PAWS_OA_cat_intake_month_counts, PAWS_OA_dog_intake
df_PAWS_OA_intake_month_counts_species <- df_PAWS_OA_intake_month_counts_species[-3]
names(df_PAWS_OA_intake_month_counts_species)[1] <- "Month"
names(df_PAWS_OA_intake_month_counts_species)[2] <- "Cats"
names(df_PAWS_OA_intake_month_counts_species)[3] <- "Dogs"

df_PAWS_OA_intake_month_counts_species <- melt(df_PAWS_OA_intake_month_counts_species, id.vars='Month')
df_PAWS_OA_intake_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), re

#Grays Ferry Avenue
GFA_cat_release_month_counts <- table(GFA_cat$release_month_fac)
GFA_dog_release_month_counts <- table(GFA_dog$release_month_fac)
df_GFA_release_month_counts_species <- data.frame(GFA_cat_release_month_counts, GFA_dog_release_month_co
df_GFA_release_month_counts_species <- df_GFA_release_month_counts_species[-3]
names(df_GFA_release_month_counts_species)[1] <- "Month"
names(df_GFA_release_month_counts_species)[2] <- "Cats"
names(df_GFA_release_month_counts_species)[3] <- "Dogs"

df_GFA_release_month_counts_species <- melt(df_GFA_release_month_counts_species, id.vars='Month')
df_GFA_release_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep(


#PAC
PAC_cat_release_month_counts <- table(PAC_cat$release_month_fac)
PAC_dog_release_month_counts <- table(PAC_dog$release_month_fac)
df_PAC_release_month_counts_species <- data.frame(PAC_cat_release_month_counts, PAC_dog_release_month_co
df_PAC_release_month_counts_species <- df_PAC_release_month_counts_species[-3]
names(df_PAC_release_month_counts_species)[1] <- "Month"
names(df_PAC_release_month_counts_species)[2] <- "Cats"
names(df_PAC_release_month_counts_species)[3] <- "Dogs"

df_PAC_release_month_counts_species <- melt(df_PAC_release_month_counts_species, id.vars='Month')
df_PAC_release_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep(

#Grant Avenue
GA_cat_release_month_counts <- table(GA_cat$release_month_fac)
GA_dog_release_month_counts <- table(GA_dog$release_month_fac)
df_GA_release_month_counts_species <- data.frame(GA_cat_release_month_counts, GA_dog_release_month_count
df_GA_release_month_counts_species <- df_GA_release_month_counts_species[-3]
names(df_GA_release_month_counts_species)[1] <- "Month"
names(df_GA_release_month_counts_species)[2] <- "Cats"
names(df_GA_release_month_counts_species)[3] <- "Dogs"

df_GA_release_month_counts_species <- melt(df_GA_release_month_counts_species, id.vars='Month')
df_GA_release_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), rep("
```

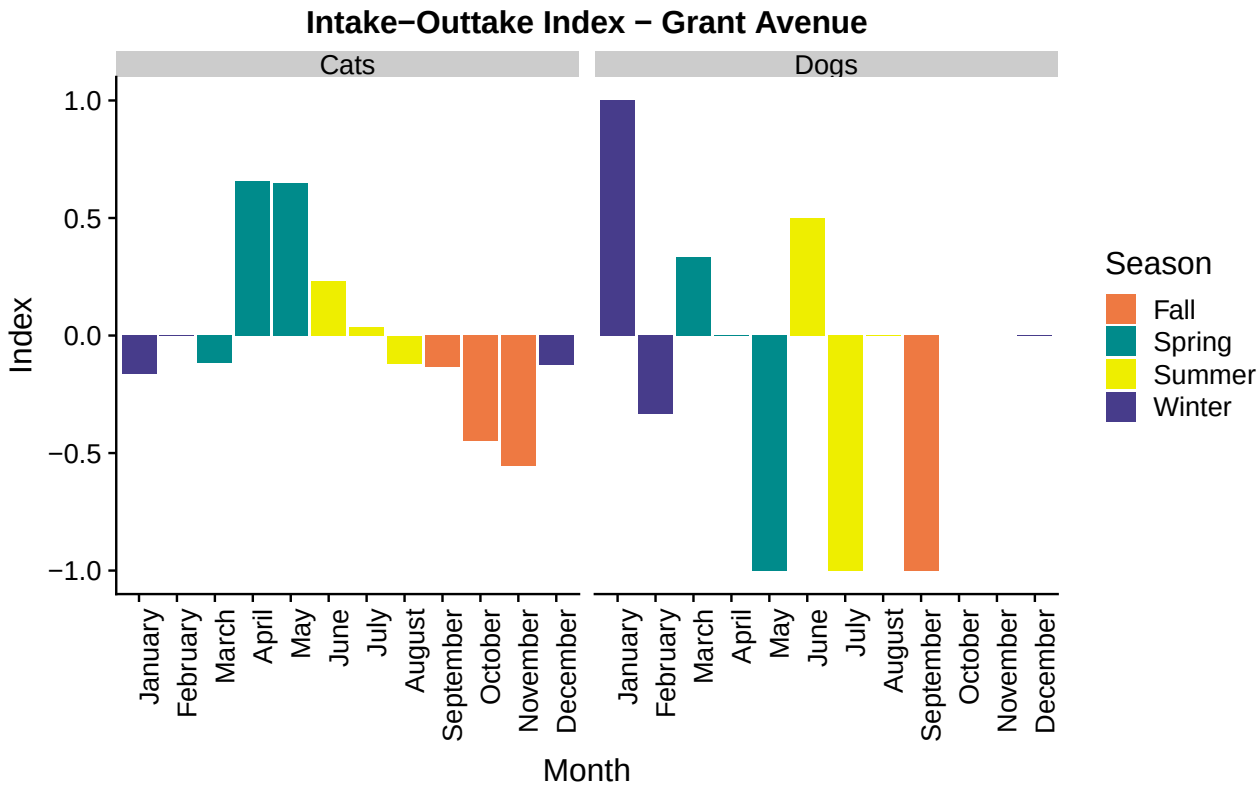
```r
#PAWS Foster Program
PAWS_FP_cat_release_month_counts <- table(PAWS_FP_cat$release_month_fac)
PAWS_FP_dog_release_month_counts <- table(PAWS_FP_dog$release_month_fac)
df_PAWS_FP_release_month_counts_species <- data.frame(PAWS_FP_cat_release_month_counts, PAWS_FP_dog_rel
df_PAWS_FP_release_month_counts_species <- df_PAWS_FP_release_month_counts_species[-3]
names(df_PAWS_FP_release_month_counts_species)[1] <- "Month"
names(df_PAWS_FP_release_month_counts_species)[2] <- "Cats"
names(df_PAWS_FP_release_month_counts_species)[3] <- "Dogs"

df_PAWS_FP_release_month_counts_species <- melt(df_PAWS_FP_release_month_counts_species, id.vars='Month
df_PAWS_FP_release_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), r


#PAWS Offsite Adoptions
PAWS_OA_cat_release_month_counts <- table(PAWS_OA_cat$release_month_fac)
PAWS_OA_dog_release_month_counts <- table(PAWS_OA_dog$release_month_fac)
df_PAWS_OA_release_month_counts_species <- data.frame(PAWS_OA_cat_release_month_counts, PAWS_OA_dog_rel
df_PAWS_OA_release_month_counts_species <- df_PAWS_OA_release_month_counts_species[-3]
names(df_PAWS_OA_release_month_counts_species)[1] <- "Month"
names(df_PAWS_OA_release_month_counts_species)[2] <- "Cats"
names(df_PAWS_OA_release_month_counts_species)[3] <- "Dogs"

df_PAWS_OA_release_month_counts_species <- melt(df_PAWS_OA_release_month_counts_species, id.vars='Month
df_PAWS_OA_release_month_counts_species$Season <- c(rep("Winter",2), rep("Spring",3), rep("Summer",3), r


#Grays Ferry Avenue
df_GFA_intake_outtake_index_species <- cbind(df_GFA_intake_month_counts_species[1],df_GFA_intake_month_

df_GFA_intake_outtake_index_species$Index <-(df_GFA_intake_month_counts_species$value-df_GFA_release_mon

ggplot(df_GFA_intake_outtake_index_species, aes(x=Month, y=Index, fill = Season)) + geom_bar(stat='ident
```



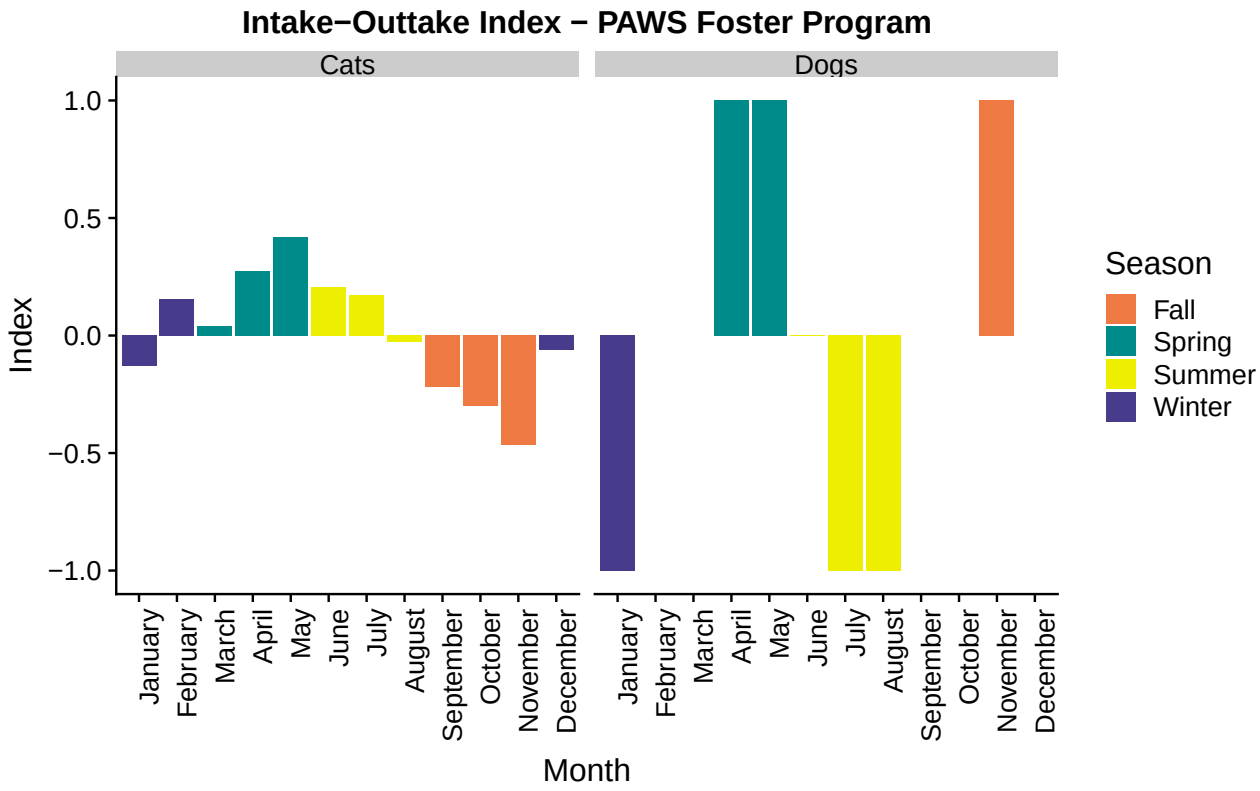

```
#PAC
df_PAC_intake_outtake_index_species <- cbind(df_PAC_intake_month_counts_species[1],df_PAC_intake_month_

df_PAC_intake_outtake_index_species$Index <-(df_PAC_intake_month_counts_species$value-df_PAC_release_mon

ggplot(df_PAC_intake_outtake_index_species, aes(x=Month, y=Index, fill = Season)) + geom_bar(stat='ident
```



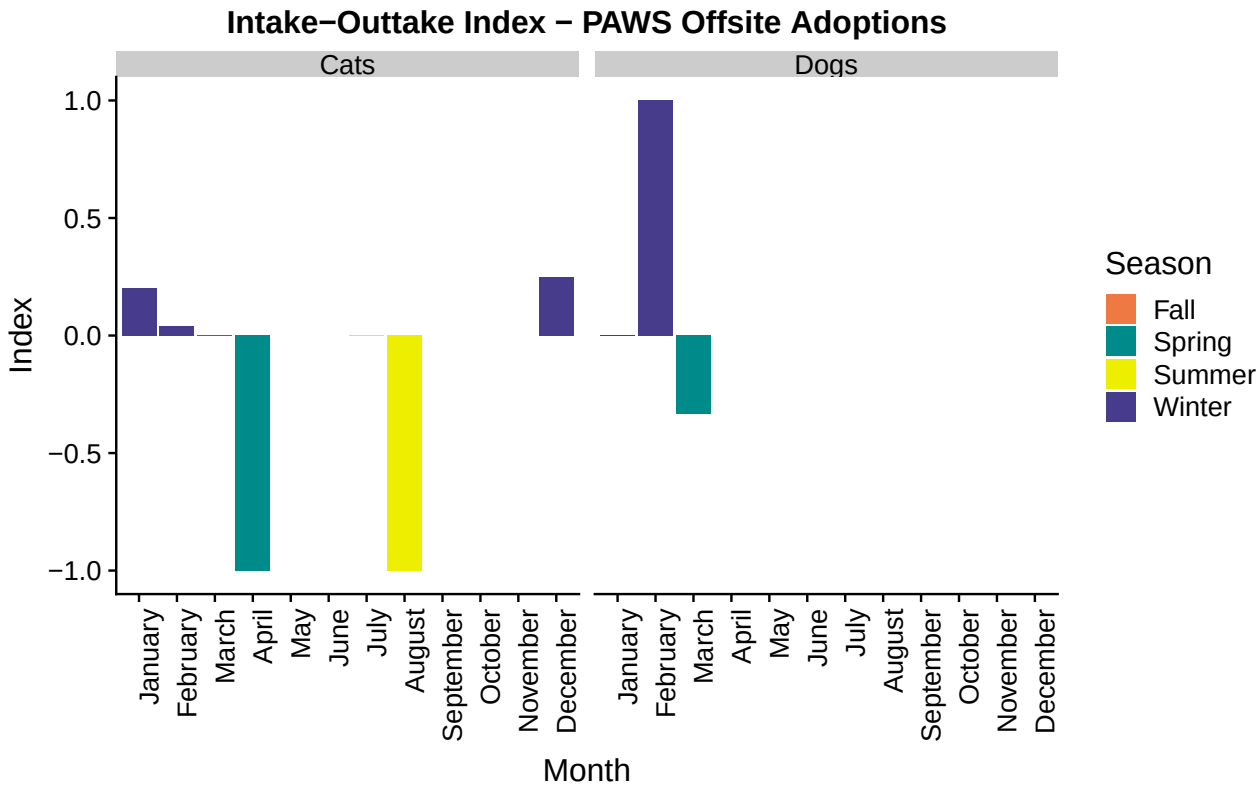

```
#Grant Avenue
df_GA_intake_outtake_index_species <- cbind(df_GA_intake_month_counts_species[1],df_GA_intake_month_cou

df_GA_intake_outtake_index_species$Index <-(df_GA_intake_month_counts_species$value-df_GA_release_month_

ggplot(df_GA_intake_outtake_index_species, aes(x=Month, y=Index, fill = Season)) + geom_bar(stat='ident
```




```
#PAWS Foster Program
df_PAWS_FP_intake_outtake_index_species <- cbind(df_PAWS_FP_intake_month_counts_species[1],df_PAWS_FP_i

df_PAWS_FP_intake_outtake_index_species$Index <-(df_PAWS_FP_intake_month_counts_species$value-df_PAWS_F

ggplot(df_PAWS_FP_intake_outtake_index_species, aes(x=Month, y=Index, fill = Season)) + geom_bar(stat='
```

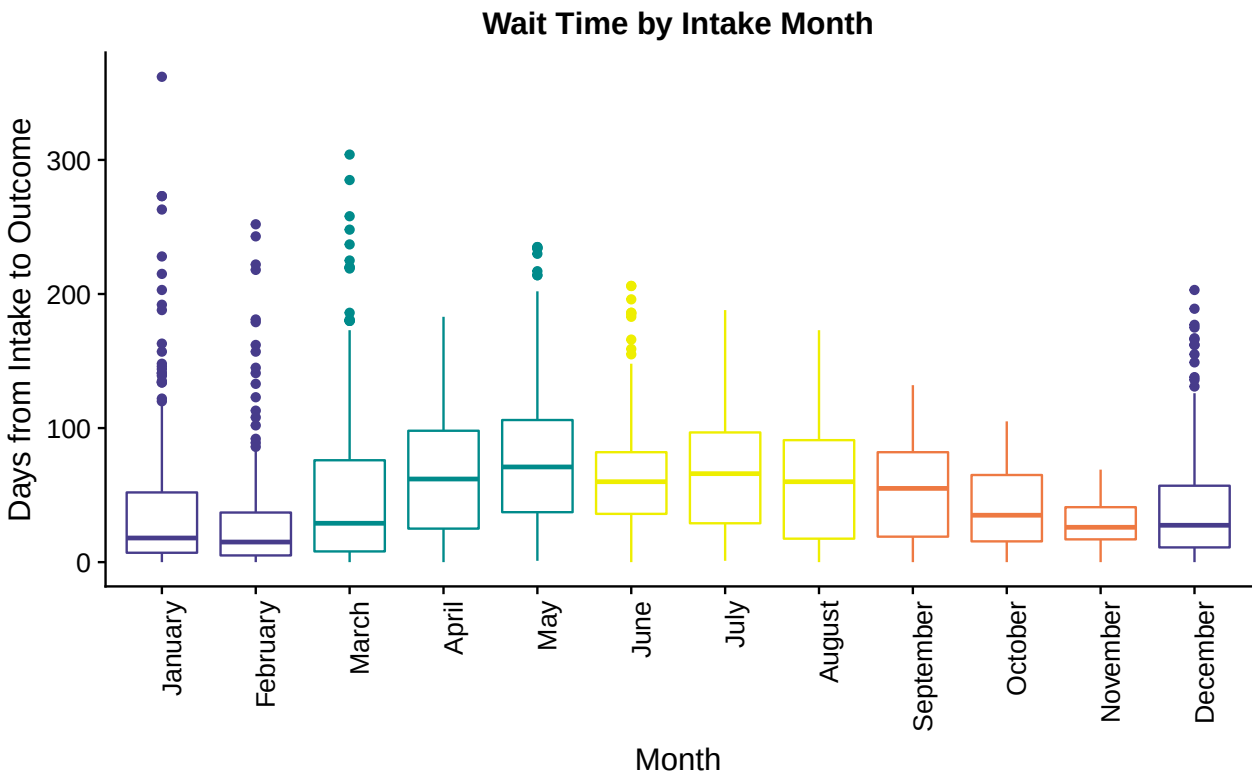



```
#PAWS Offsite Adoptions
df_PAWS_OA_intake_outtake_index_species <- cbind(df_PAWS_OA_intake_month_counts_species[1],df_PAWS_OA_i

df_PAWS_OA_intake_outtake_index_species$Index <-(df_PAWS_OA_intake_month_counts_species$value-df_PAWS_O

ggplot(df_PAWS_OA_intake_outtake_index_species, aes(x=Month, y=Index, fill = Season)) + geom_bar(stat='
```



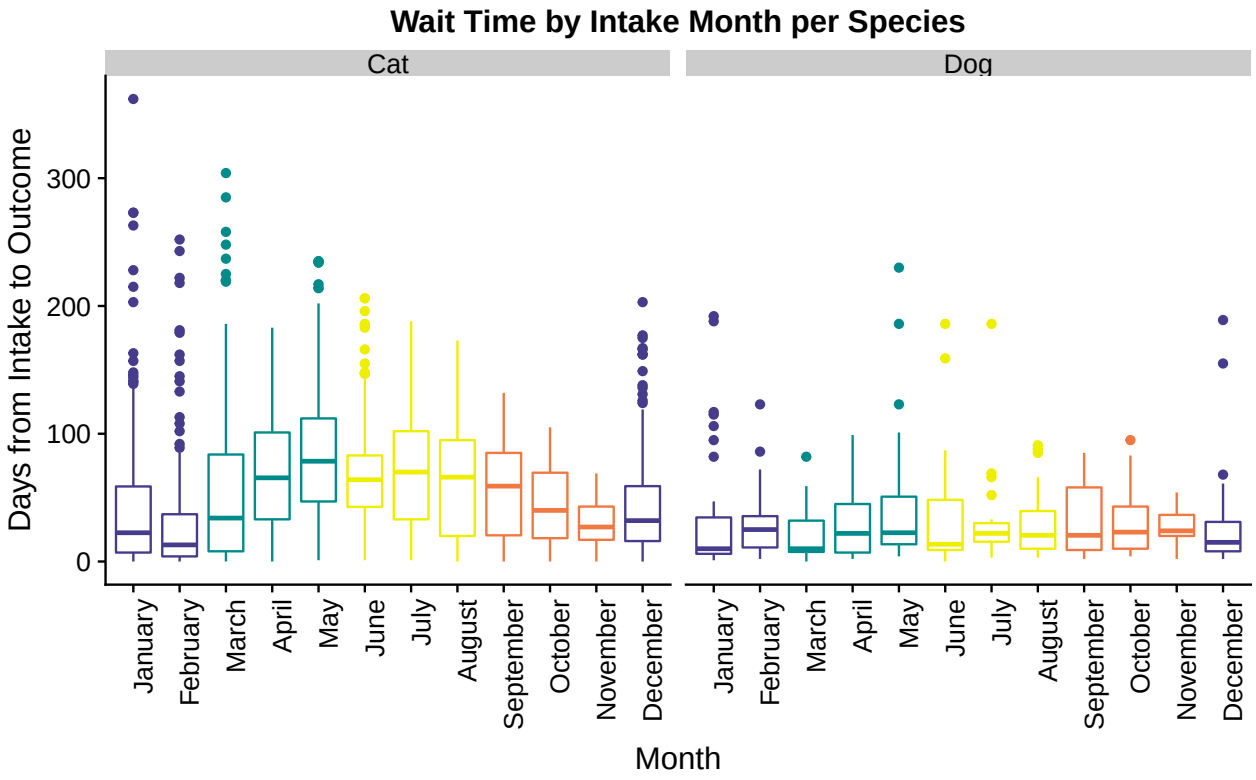
We then examined seasonal patterns in wait time:

```
cols = c("January" = "slateblue4", "February" = "slateblue4", "March" = "cyan4", "April" = "cyan4", "Ma

ggplot(master_animal, aes(x=intake_month_fac, y=wait_days, col = intake_month_fac))+
    geom_boxplot()+
    scale_colour_manual(values = cols)+
    ggtitle("Wait Time by Intake Month") +
    theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle=90, hjust=1), legend
        ylab("Days from Intake to Outcome")+
        xlab("Month")
```



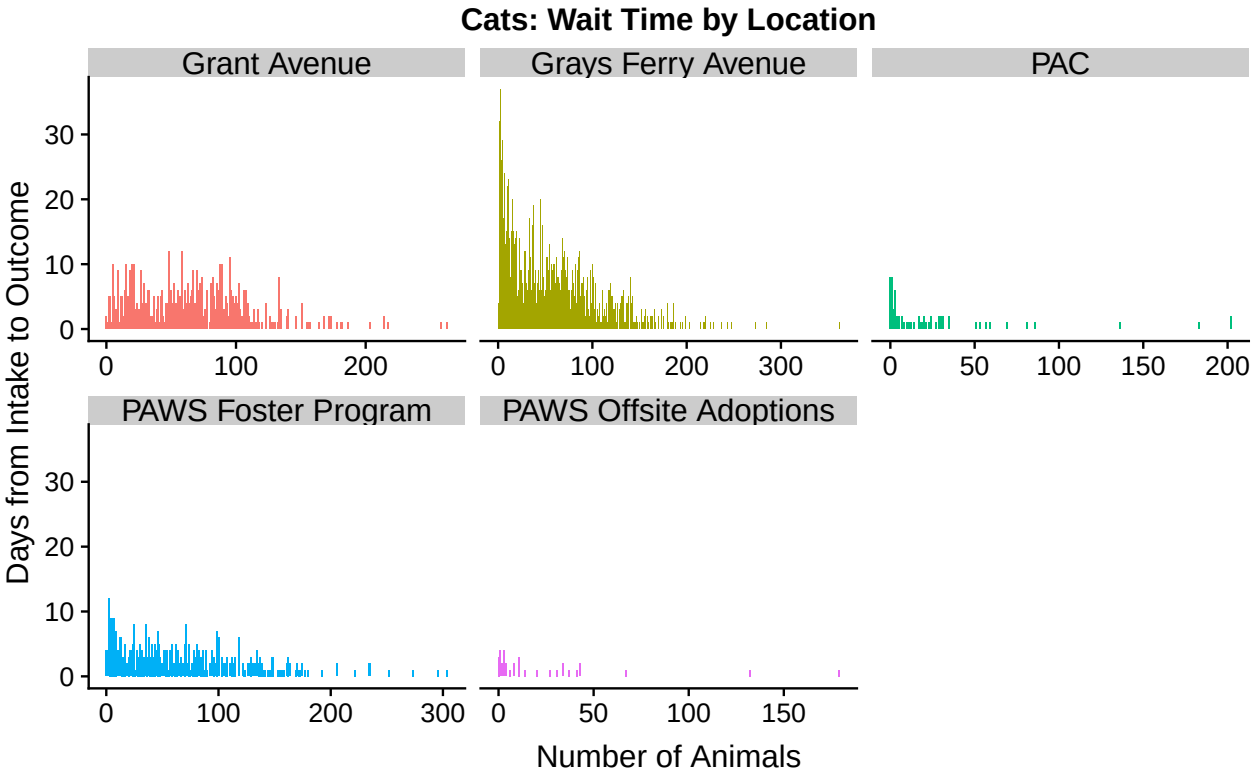
And examined seasonal patterns in wait time per species:

```
ggplot(master_animal, aes(x=intake_month_fac, y=wait_days, col = intake_month_fac))+
    geom_boxplot()+
    scale_colour_manual(values = cols)+
    facet_wrap(~species)+
    ggtitle("Wait Time by Intake Month per Species") +
    theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle=90, hjust=1), legend
        ylab("Days from Intake to Outcome")+
        xlab("Month")
```



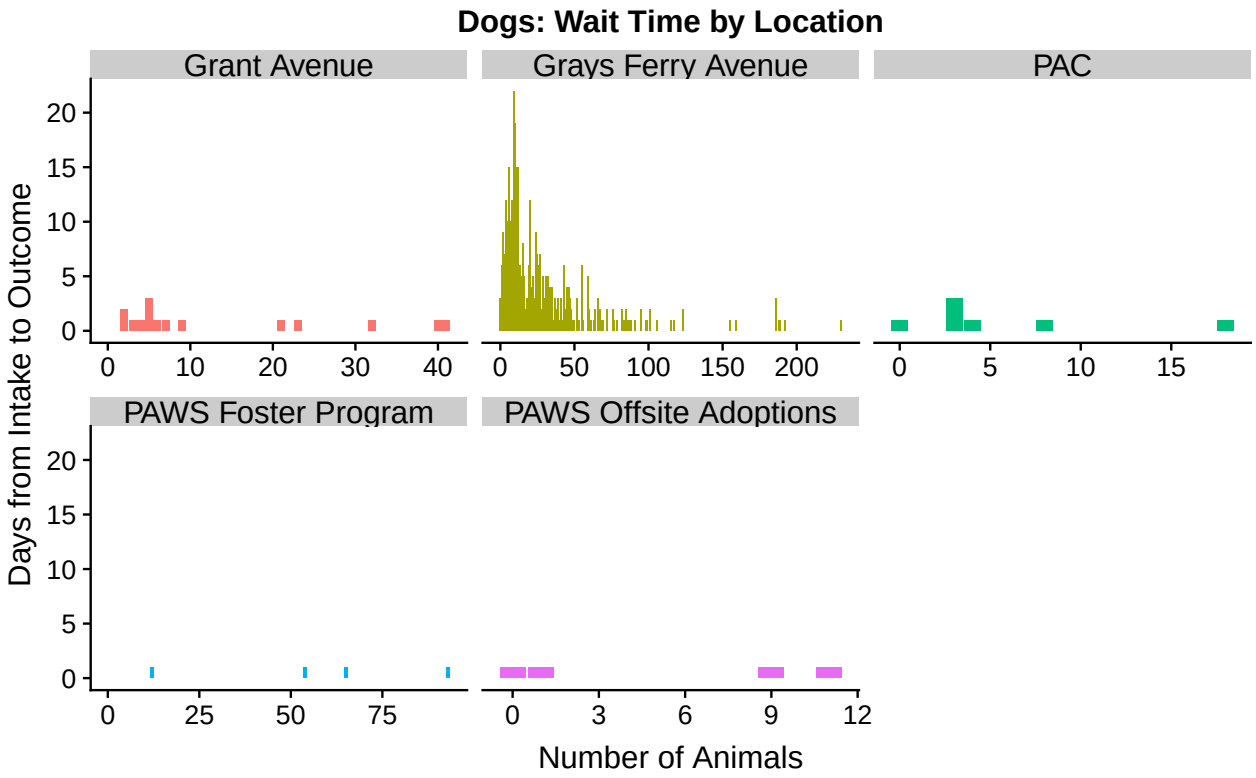
We then visualized the wait time by location for each species.

```
raw_data %>%
    filter(species == "Cat") %>%
    filter(!is.na(intake_sitename)) %>%
    group_by(intake_sitename) %>%
    group_by(wait_days, intake_sitename) %>%
    summarise(n=n()) %>%
    ggplot(aes(x = wait_days, y = n, fill = intake_sitename)) +
        geom_col() +
        facet_wrap(~intake_sitename, scales = "free_x") +
        expand_limits(x = 0, y = 0) +
        annotate("text", label= median) +
        xlab("Days from Intake to Outcome") +
        theme(legend.position = "none", strip.text = element_text(size=14))+
          xlab('Number of Animals') +
        ylab('Days from Intake to Outcome')+
        ggtitle("Cats: Wait Time by Location")
```



Our findings demonstrated that PAWS offsite adoptions has the shortest wait time for cats.

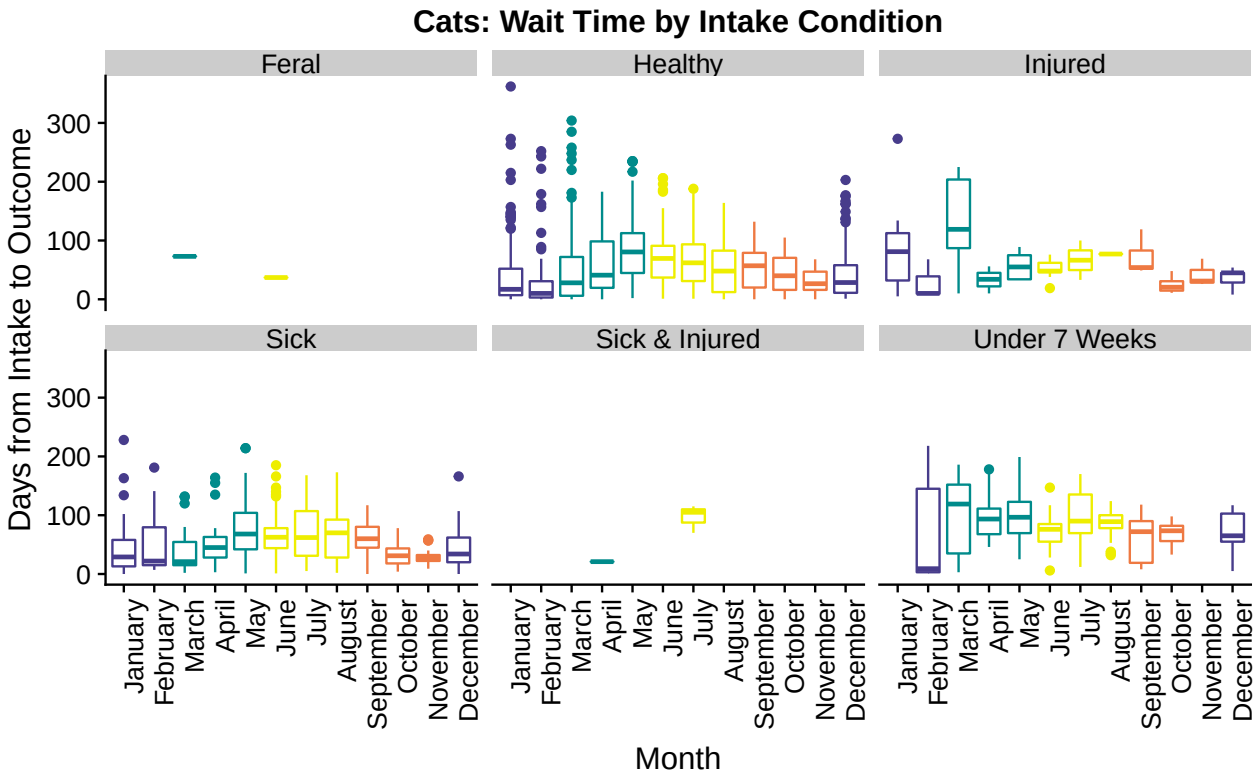
```
raw_data %>%
    filter(species == "Dog") %>%
    group_by(intake_sitename) %>%
    group_by(wait_days, intake_sitename) %>%
    summarise(n=n()) %>%
    ggplot(aes(x = wait_days, y = n, fill = intake_sitename)) +
        geom_col() +
        facet_wrap(~intake_sitename, scales = "free_x") +
        expand_limits(x = 0, y = 0) +
        xlab("Days from Intake to Outcome") +
        theme(legend.position = "none", strip.text = element_text(size=14))+
        xlab('Number of Animals') +
        ylab('Days from Intake to Outcome')+
        ggtitle("Dogs: Wait Time by Location")
```



Our findings demonstrated that the PAC location has the shortest wait time for dogs.

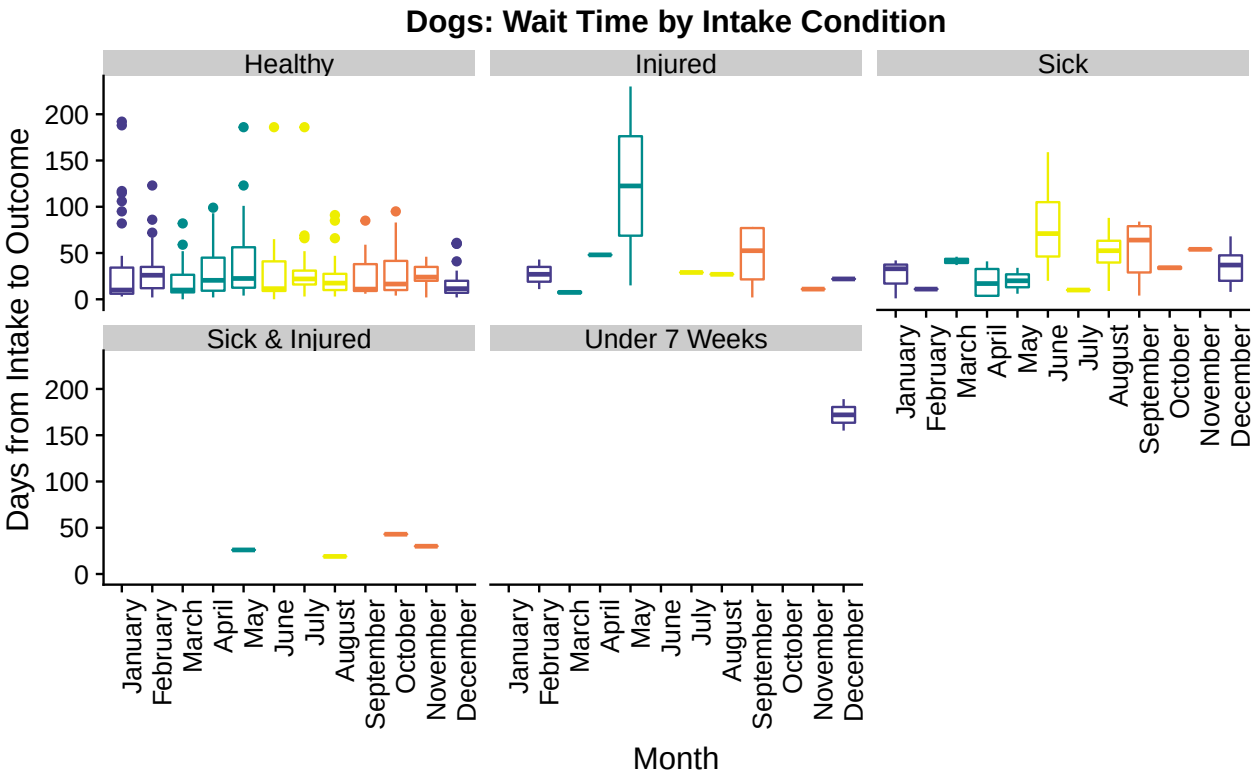
Last, we examined wait time by intake health condition for each species.

```
master_animal %>%
    filter(species=="Cat") %>%
    ggplot(., aes(x=intake_month_fac, y=wait_days, col = intake_month_fac))+
    geom_boxplot()+
    scale_colour_manual(values = cols)+
    facet_wrap(~intake_condition)+
    ggtitle("Cats: Wait Time by Intake Condition")+
    theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle=90, hjust=1), legend
        ylab("Days from Intake to Outcome")+
        xlab("Month")
```



For cats, we found a that animals classified as 'Sick', 'Injured', and 'Under 7 Weeks' tended to have longer wait times in the spring and summer months.

```
master_animal %>%
    filter(species=="Dog") %>%
    ggplot(., aes(x=intake_month_fac, y=wait_days, col = intake_month_fac))+
    geom_boxplot()+
    scale_colour_manual(values = cols)+
    facet_wrap(~intake_condition)+
    ggtitle("Dogs: Wait Time by Intake Condition")+
    theme(plot.title = element_text(hjust = 0.5), axis.text.x = element_text(angle=90, hjust=1), legend
        ylab("Days from Intake to Outcome")+
        xlab("Month")
```



For dogs, we found a that animals classified as 'Injured' and 'Sick' also tended to have longer wait times in the spring and summer months.

Conclusions and Next Steps

In conclusion, our analyses revealed that animal characteristics, intake characteristics, and seasonal and locational patterns contribute to animals' wait times at PAWS in the 2018 year. Almost all animals' outcomes from 2018 were adoptions, meaning that PAWS is fulfilling its goal of finding homes for needy animals in the Philadelphia area. The median wait time for cats (51 days) was longer than the median wait time for dogs (18 days), and this was likely due to 1) greater number of cats vs dogs, and 2) longer wait times for sick, young cats in the spring and summer months. We also observed differences in wait time by PAWS location for each species, but this is likely due to the number and species of animals at each location. Overall, our findings indicate that PAWS may want to focus resources on young and sick cats in the spring and summer months in order to reduce wait times.

2. Application Trajectories

Contributors

- **Ramaa Nathan** (group leader) is an aspiring data scientist with a PhD in Computer Science and an ongoing masters in Applied Statistics. Her background is in finance and healthcare.
- **Kate Connolly** is a digital analyst at the Philadelphia Inquirer where she helps to maintain the analytics framework and to provide data-driven support and decisions across the organization.
- **Veena Dali** is a senior business intelligence analyst at Comcast working to provide data solutions to support business decisions. Her background is in Neuroscience and Computer Science.
- **Amy Goodwin Davies** is a data scientist with a background in psycholinguistics.
- **Brendan Graham** is a clinical data analyst at The Children's Hospital of Philadelphia with a background in applied statistics.
- **Ambika Sowmyan** heads the Marketing data analytics group at Hartford Funds. Her background is in Finance and Retail and has a graduate degree in Management and Predictive Analytics.

Summary

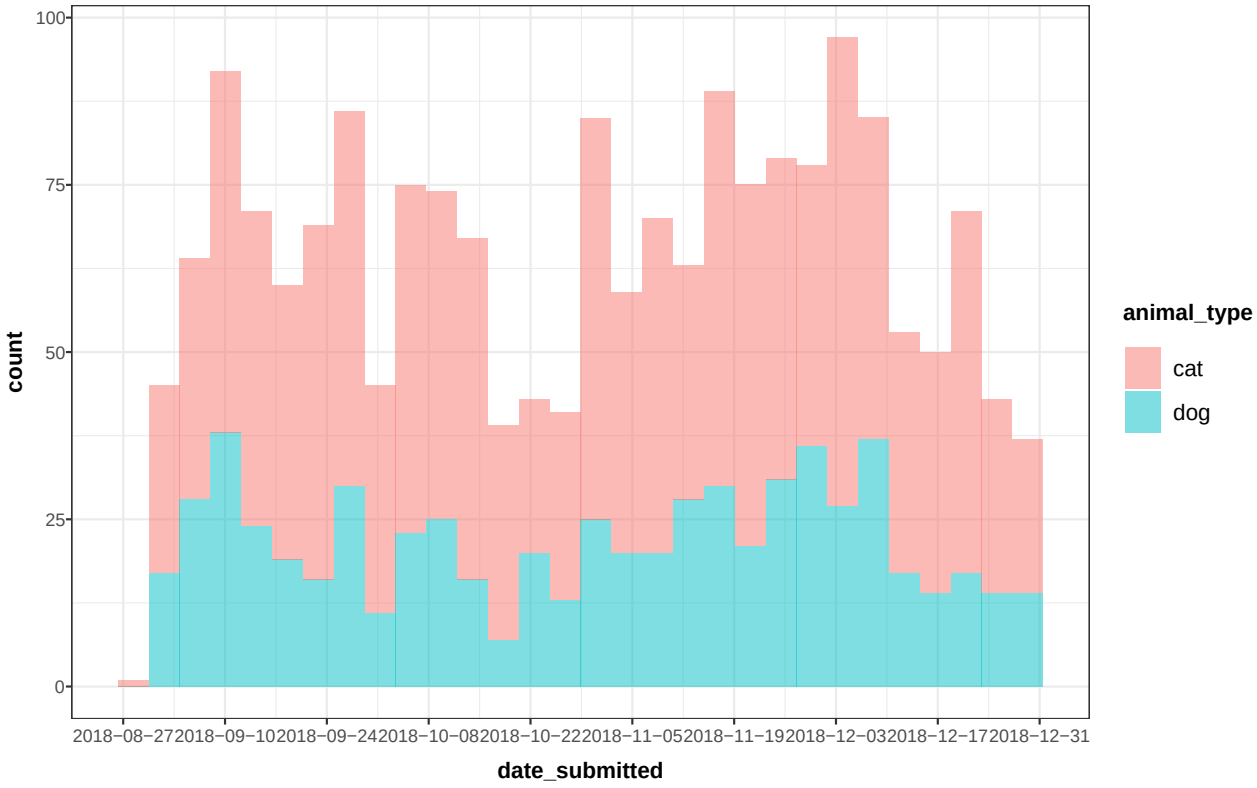
- The cleaned dataset we used for our analysis contained 1594 unique trello IDs with application submitted dates between 2018-08-30 to 2018-12-31
- For applications that resulted in adoption, cat applications took longer to process than dog applications. Cat apps took 19 days and dog apps took 8 days on average.
- Based on the animal's outcome site, adoption times were faster at PAWS Foster Program & PAWS Offsite Adoptions locations.
- Singles seem to prefer to adopt a pet.
- There were 12 denied applications and 133 red flagged
- For the denied applications, the applicants had no known allergies and many of them had unfortunate incidents with prior pets
- There was a lot of missing data in the applications especially for the home pet policy question.
- We recommend redesiging the application to enforce standardized, limited, and logical responses. For example, for certain questions, allow only a single response or provide a drop down menu. Doing so will save PAWS staff time when reviewing applications and help with future analyses.

Data Pre-processing

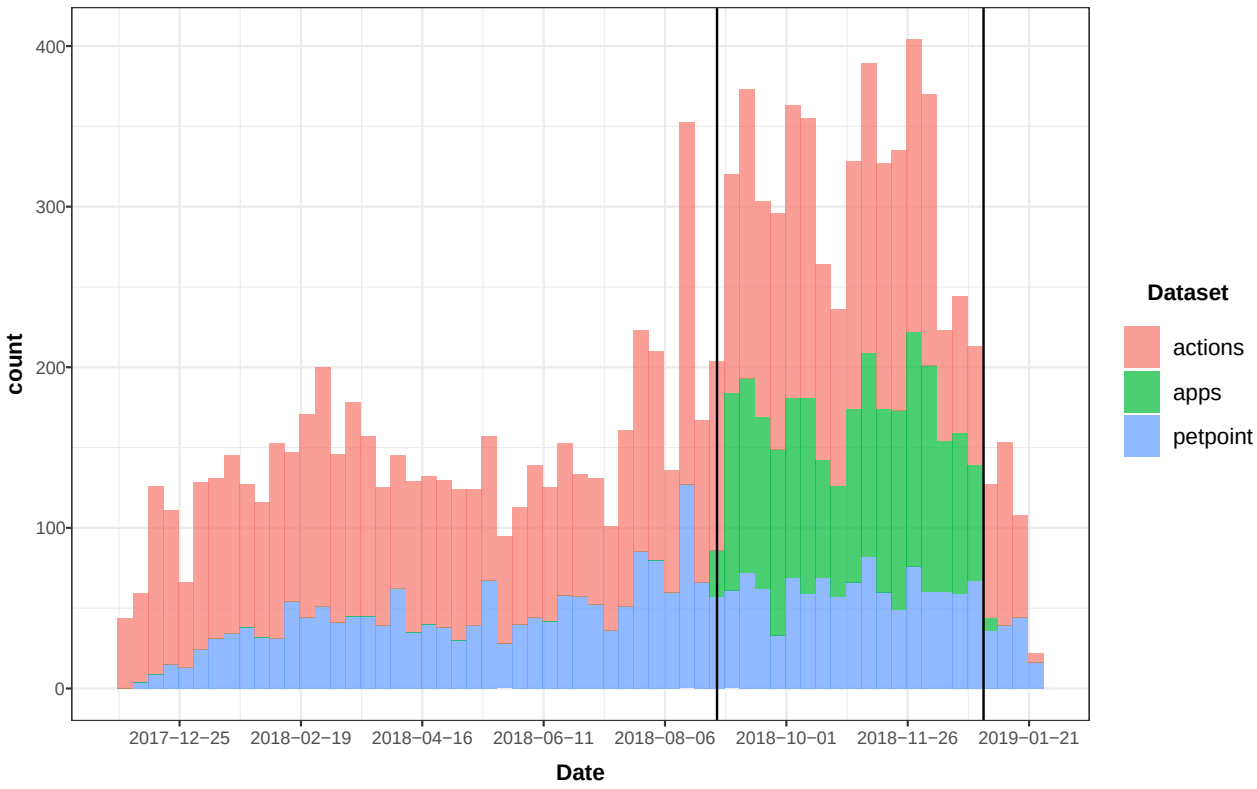
As our group focussed on questions about application trajectories, our starting point was an applications dataset comprised of `dog_apps.csv` and `cat_apps.csv`. For data pre-processing, the following steps were particularly important:

- Standardizing responses for `ideal_adoption_timeline`, `all_household_agree`, `home_pet_policy`, `experience` and `pet_kept`. For example, `ideal_adoption_timeline` had responses "next-few-weeks" and "few-weeks" which we standardised as one response ("few-weeks"). (See further discussion of this issue in the *Data Issues affecting Analyses* section)
- For `children_in_home` and `adults_in_home`, ignoring "-" by taking the absolute value and replacing absurd values with NA (we replaced values greater than 15 with NAs).
- Capping `budget_monthly` and `budget_emergency` at $10000 and $20000 respectively.
- Addressing spelling variations in the `City` variable. For example, replacing the strings "PHILLY", "FILADELFIA", "PHILIDELPHIA", "PHIMADELPHIA", "PHIALADELPHIA", "PHIALDELPHIA", "PHILDELPHIA" with "PHILADELPHIA".
- Adding new indicator variables for variables containing lists of responses. For example, from `allergies` we created indicator variables for each response (`allergies_mildly.allergic_ind`, `allergies_no.allergies_ind`, `allergies_not.sure_ind`, `allergies_very.allergic_ind`).

Our cleaned applications dataset contained 1906 rows, 1594 unique trellos ids and the submitted dates ranged from 2018-08-30 to 2018-12-31:



To our applications dataset we added fields from the actions dataset (comprised of `dog_actions.csv` and `cat_actions.csv`), the cards dataset (comprised of `dog_cards.csv` and `cat_cards.csv`), and the petpoint dataset (`petpoint.csv`) to create our dataset for analysing successful applications. One issue we encountered was that the date range for the applications dataset (123 days) was considerable smaller than the actions and petpoint datasets (417 and 413 days respectively).



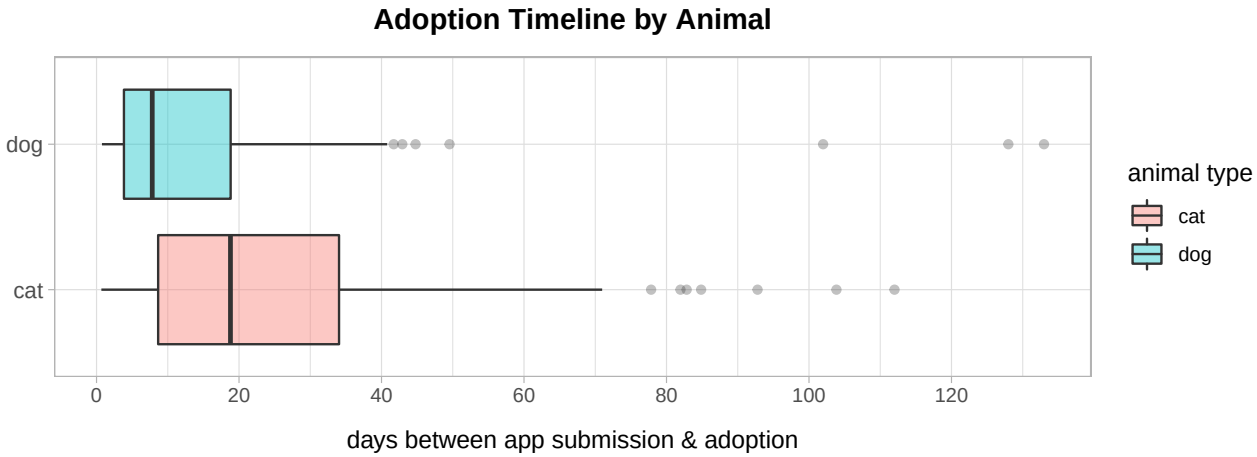
Similar data-preprocessing steps as we took for the applications dataset were taken for the actions, cards, and petpoints datasets. For the actions dataset, it was important for us to change the format of the data so each row represented a single `trello id` (prior to this each `checklist_item` for single `trello_id` had a separate row). For the purpose of analysing the denied and red-flagged applications, we created another dataset comprised of the applications and the cards datasets.

Analysis of Time in Processing Applications

How Animal & Outcome Site Influence Application Timelines

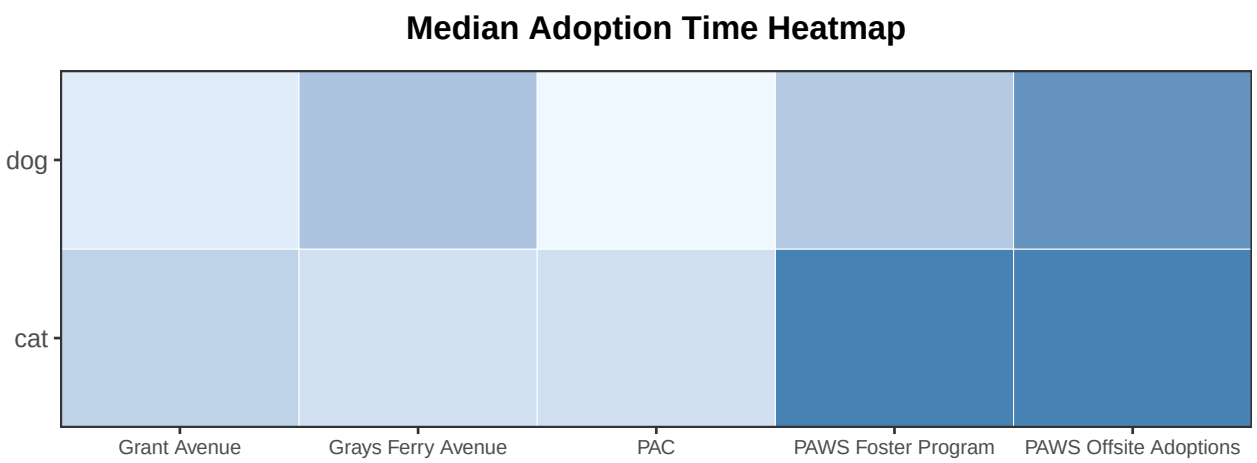
Application timelines were measured by taking the difference between the time an application was submitted and the time that application resulted in an adoption. Only applications that resulted in adoption were assessed; applications that were denied were not included in the analysis. This is a potential area of further investigation.

In general, cat applications typically take longer than dog applications. The chart below shows that the median adoption timeline for **cats** is approximately **19** days (vertical black line inside red box), while **dog** applications average about **8** to result in an adoption (vertical black line inside blue box).



The chart also illuminates that for longer-than-average application timelines, animal type may influence just *how much longer* those above-average timelines are. Of the longer-than-usual applications, cat ones took between 35 days and 70 days compared to about 18 days to 40 days for dogs.

The outcome site for an adoption also influences the timeline of an application. It's important to note that this analysis does not consider all the potential locations that an animal spent its time during the application process; it is strictly based on the animal's outcome site.



outcome_sitename	animal_type	n	median adoption time
Grant Avenue	cat	74	10
	dog	19	6
Grays Ferry Avenue	cat	2	8
	dog	18	13
PAC	cat	70	8
	dog	17	4
PAWS Foster Program	cat	187	25
	dog	20	12
PAWS Offsite Adoptions	cat	44	25
	dog	1	22

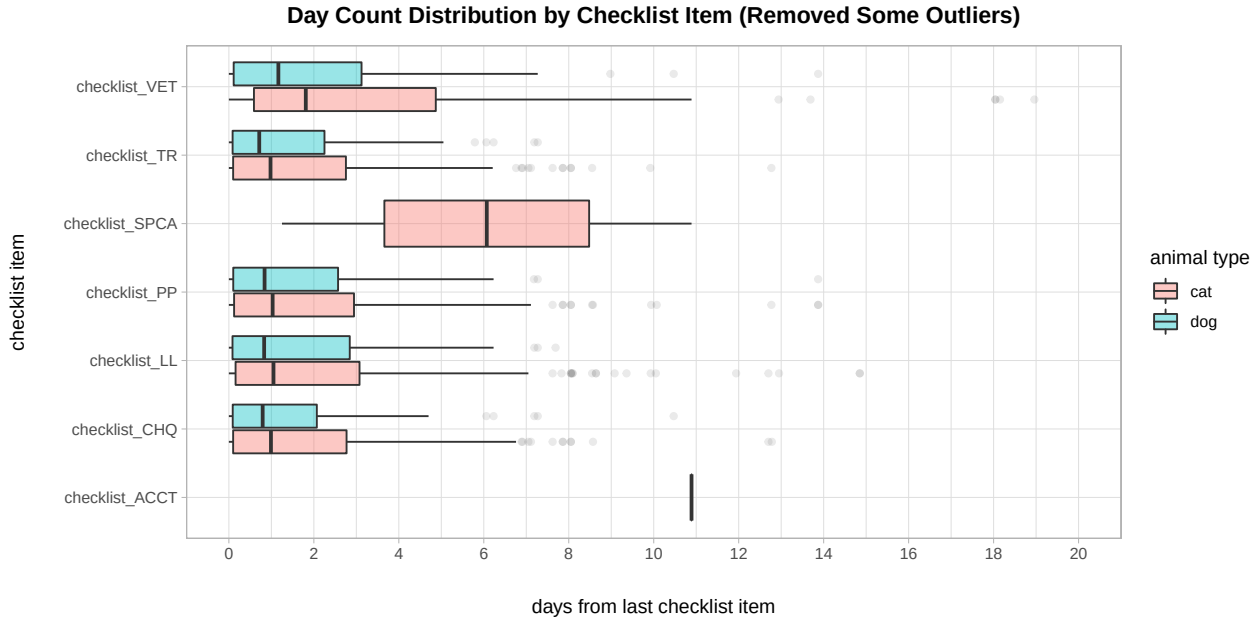
From the heatmap and table above, it's clear that overall average adoption times were higher at PAWS Foster Program & PAWS Offsite Adoptions locations. This is especially true for cat applications at those places

Based on median values, here are the fastest & slowest time-to-adoption sites:

- **Cats**
  - Slowest: PAWS Foster Program
  - Fastest: Grays Ferry Avenue
- **Dogs**
  - Slowest: PAWS Foster Program
  - Fastest: PAC

Only one site had a higher median adoption time for dogs than for cats—Grays Ferry Avenue. This site also had the fewest cat adoptions, though (n=2). It's also important to note the small n size for dog apps at PAWS Offsite Adoptions (n=1).

How Animal & Outcome Site Influence Application Checklist Items



Most application items took between one and two days (median) to complete. While the animal type and outcome site didn't significantly impact the individual item times, cat applications generally exhibited slightly longer times between checklist items. Cat applications averaged about **1.2** days between checklist item, compared to **0.9** for dogs (excluding SPCA & ACCT items). The VET checklist item had the greatest difference between cats and dogs, and also was the item that took the longest (besides SPCA & ACCT items). This distinction between animals, while modest, could contribute to longer submission-to-adoption times for cat applications.

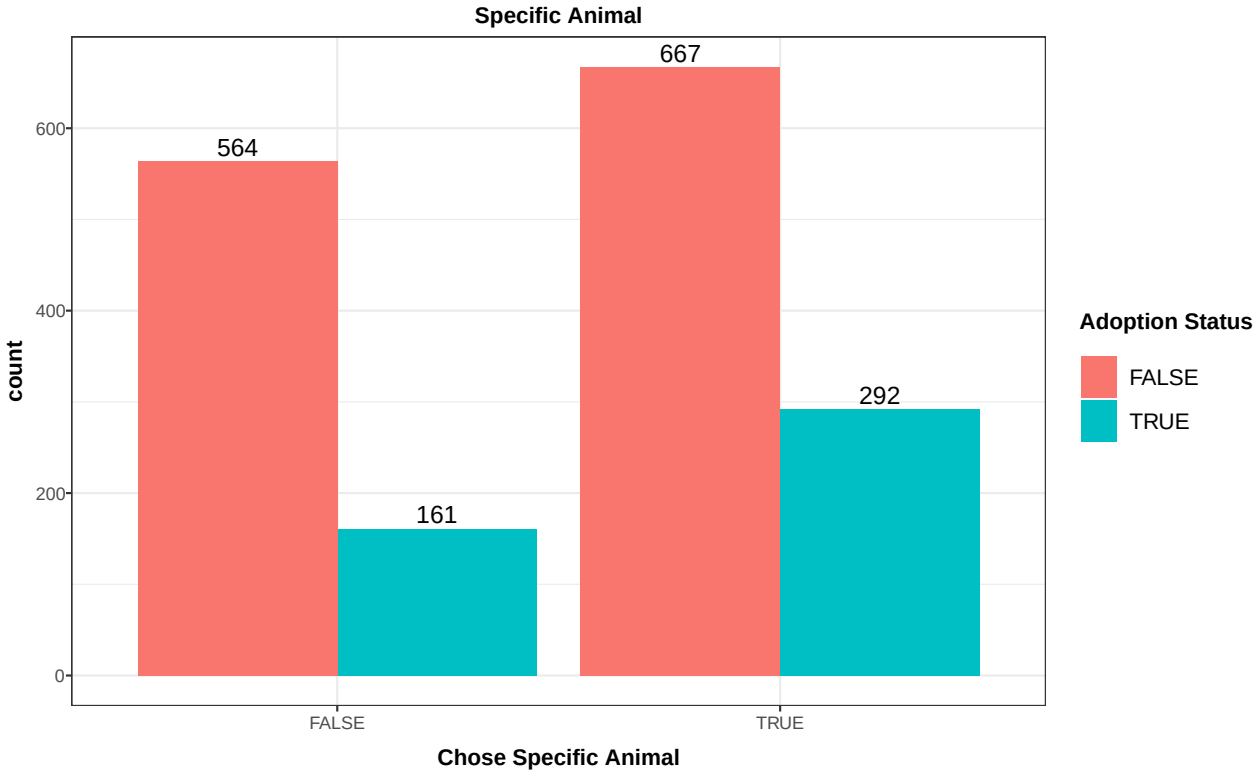
The chart above removed significant outliers, but further inspection of these outliers could be valuable. Understanding what causes certain application steps to take longer could help to streamline parts of the checklist process.

checklist item	n	median days from last item	percent of cards with item checked
checklist_ACCT	1	10.89	0.2%
checklist_SPCA	2	6.07	0.4%
checklist_VET	425	1.80	93.8%
checklist_CHQ	432	0.97	95.4%
checklist_LL	433	1.03	95.6%
checklist_PP	433	1.03	95.6%
checklist_TR	435	0.95	96.0%

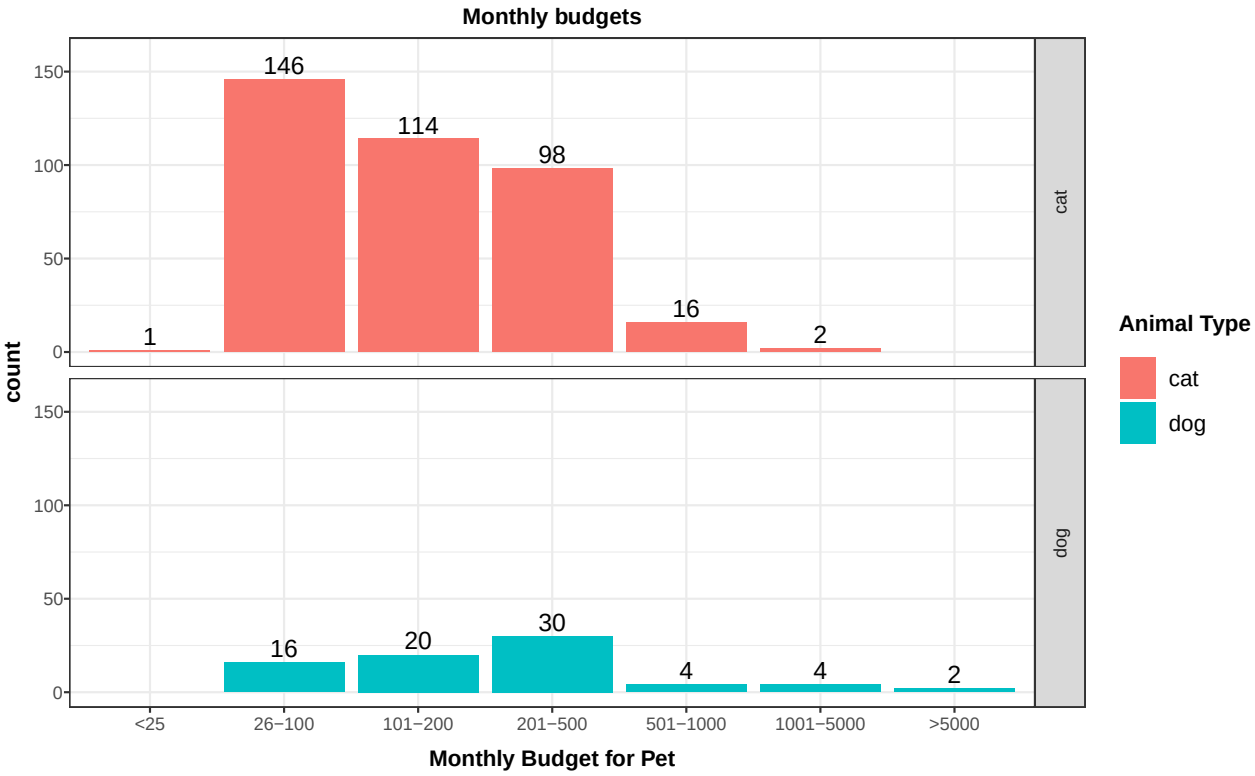
The table above shows the exceptions to the average checklist times. The ACCT and SPCA checklist items took considerably longer to complete than other items, but they also were present in less than 1% of applications. This low sample limits any sound conclusions, but does present an area for potential further exploration. It may be valuable to assess if other components of an application—like red flags or particular animal information—lead to this item being more mandatory. But more data would be needed for this analysis.

Analysis of Application Characteristics that Result in Adoption

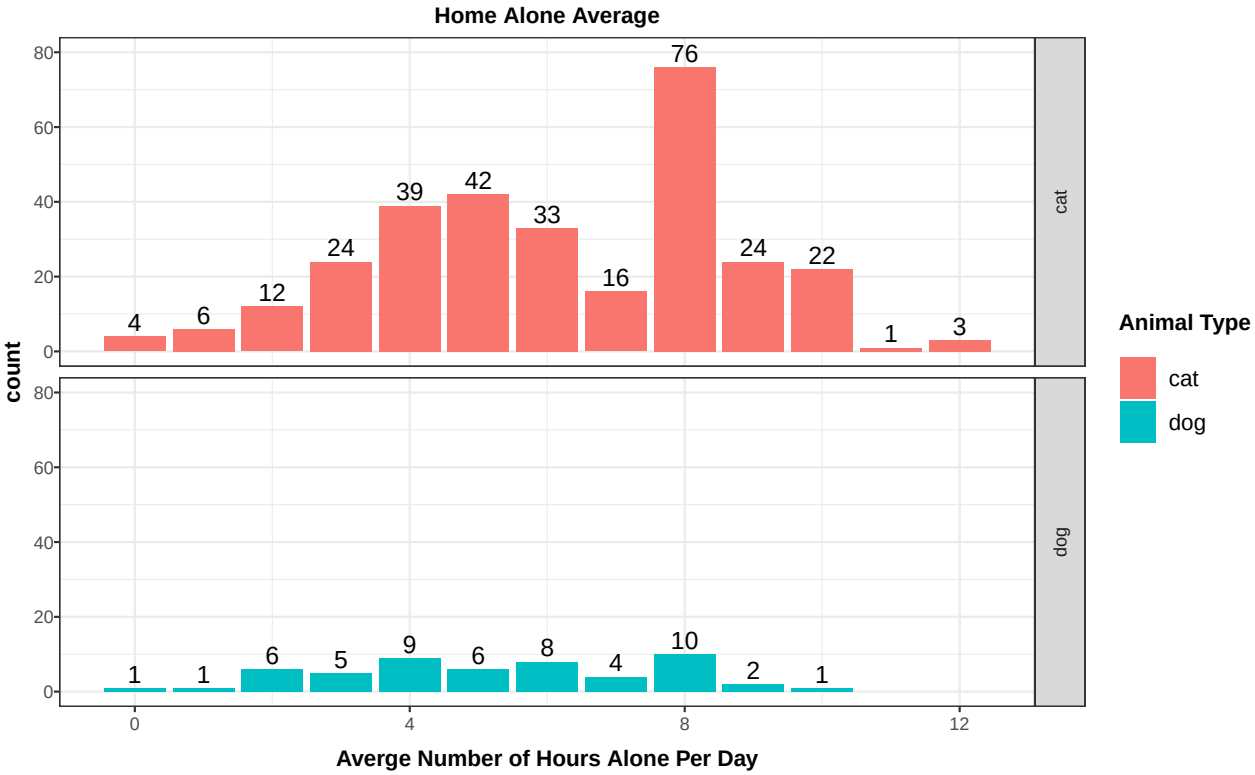
We analysed the the different factors of the applications that ended with a successful adoption.



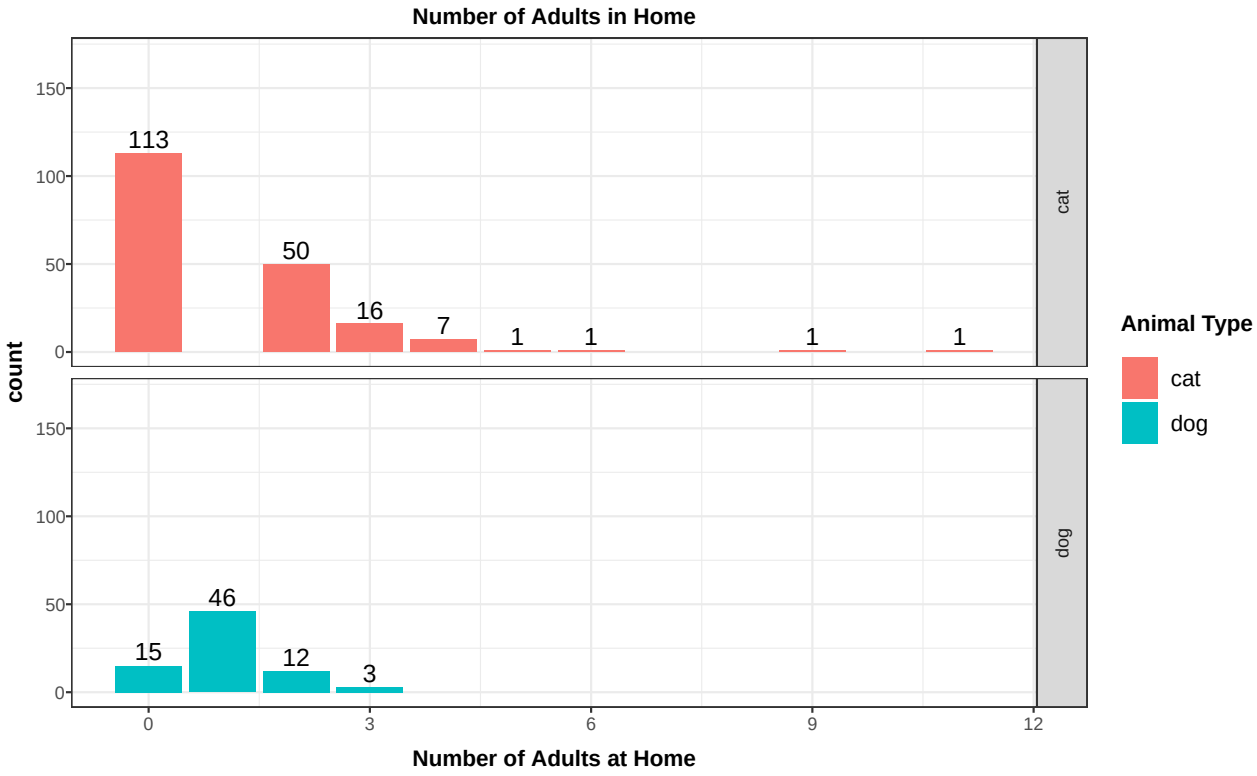
When applicants requested a specific type of animal, 30% of applications resulted in an adoption vs only 22% of the applications resultd in an adoption. This seems surprising as we would expect an applicant who is not specific about the type of animal to be able to adopt easily.



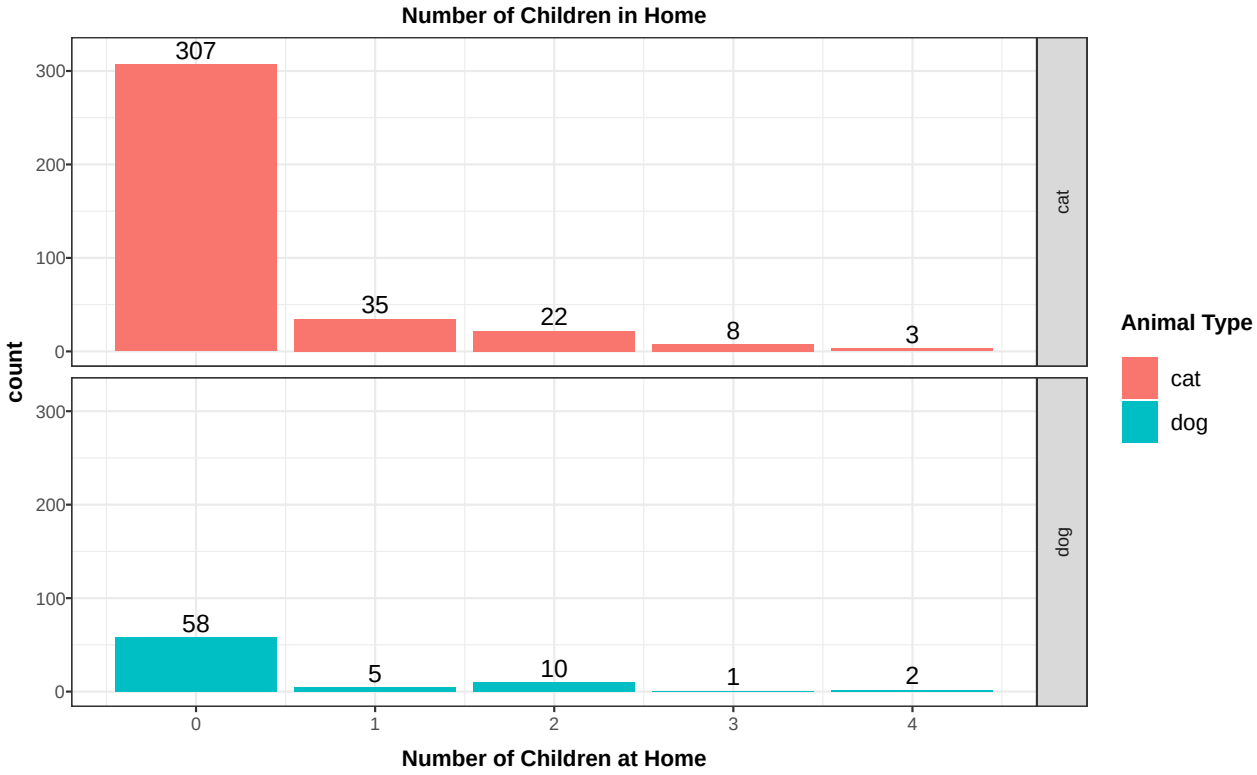
Most of the applicants who adopted a pet had allocated a monthly budget of less than $500.



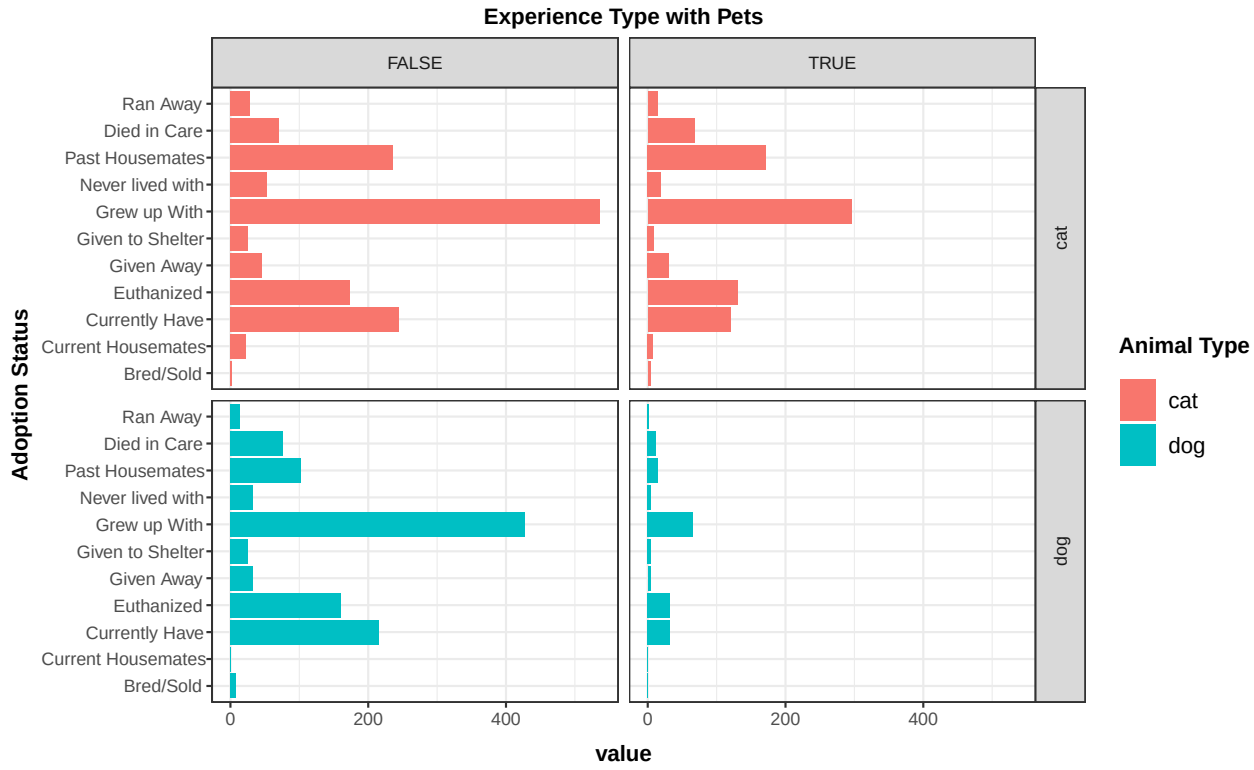
Applicants who expected to leave the animal alone at home for longer hours chose to adopt a cat. The largest number of applicants expected the animal to be alone for 8 hours, which would be typical of an applicant who works full time.



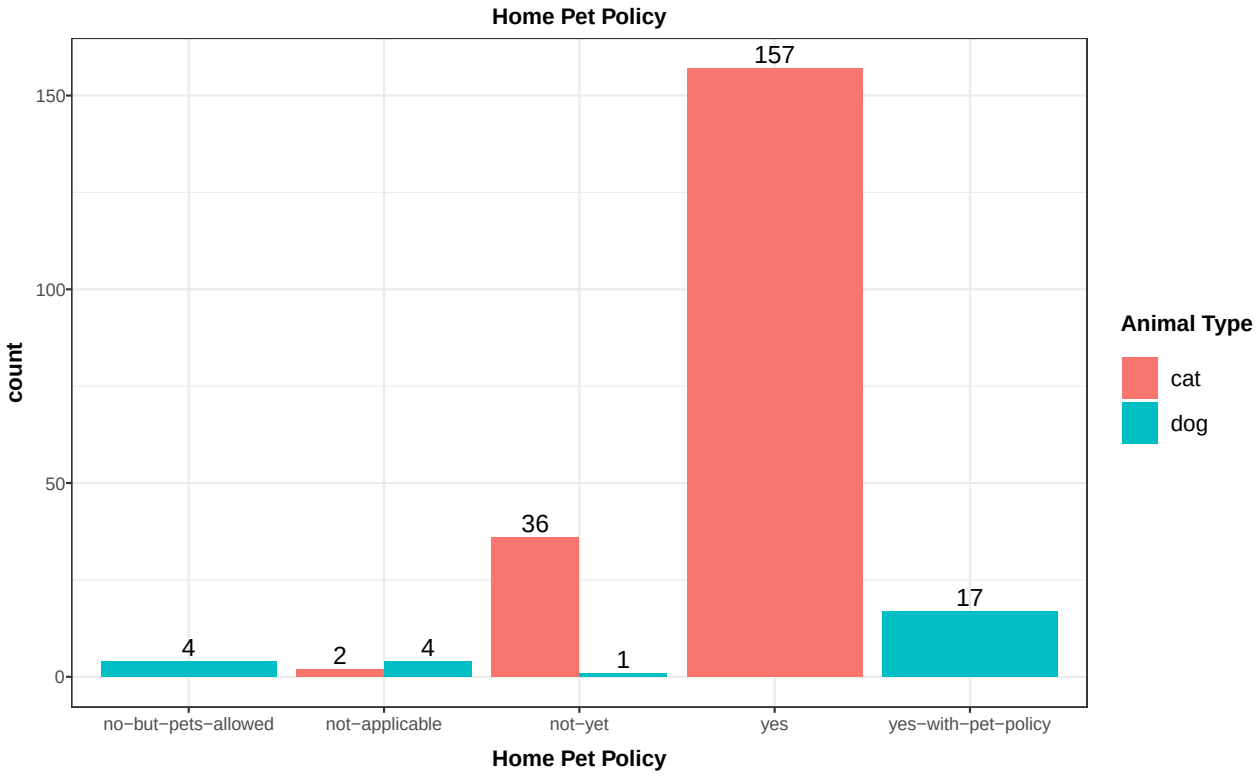
Singles overwhelmingly seem to prefer to adopt a pet.



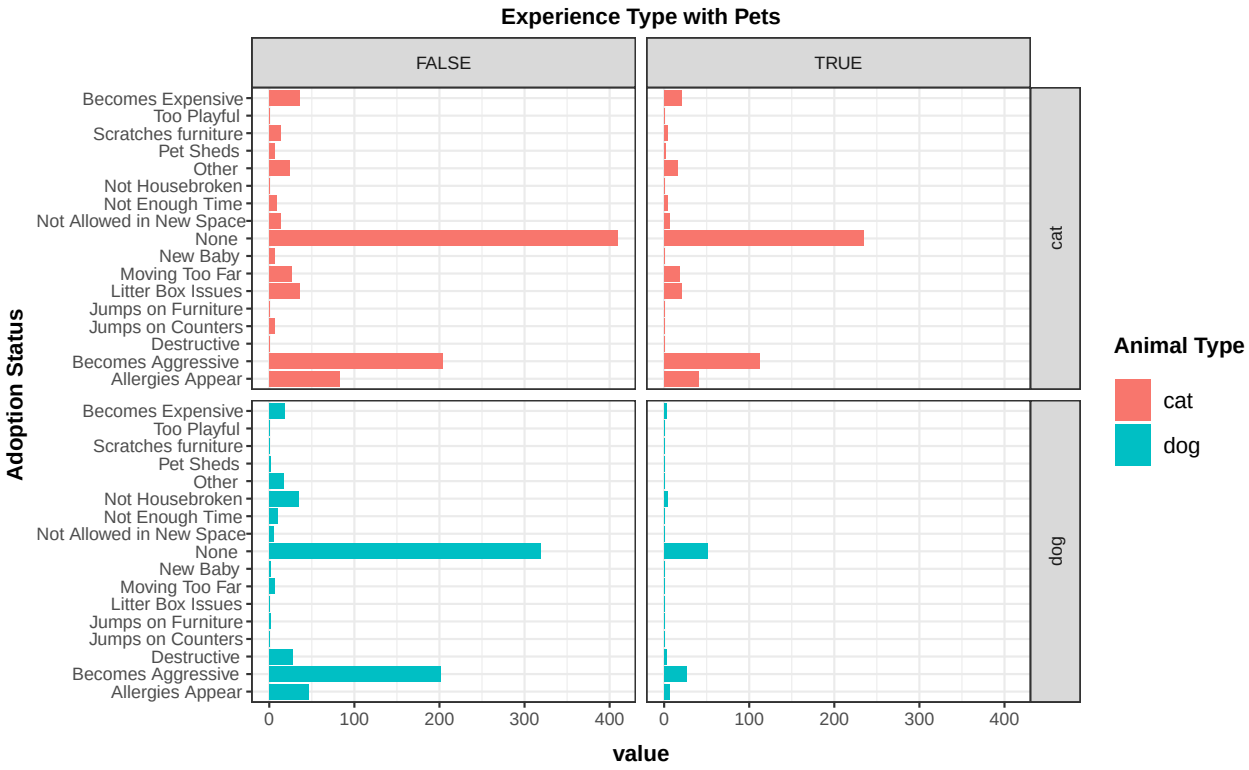
Again, families with no children at home seem to be the largest number of applicants. This correlates with mostly singles wanting to adopt.



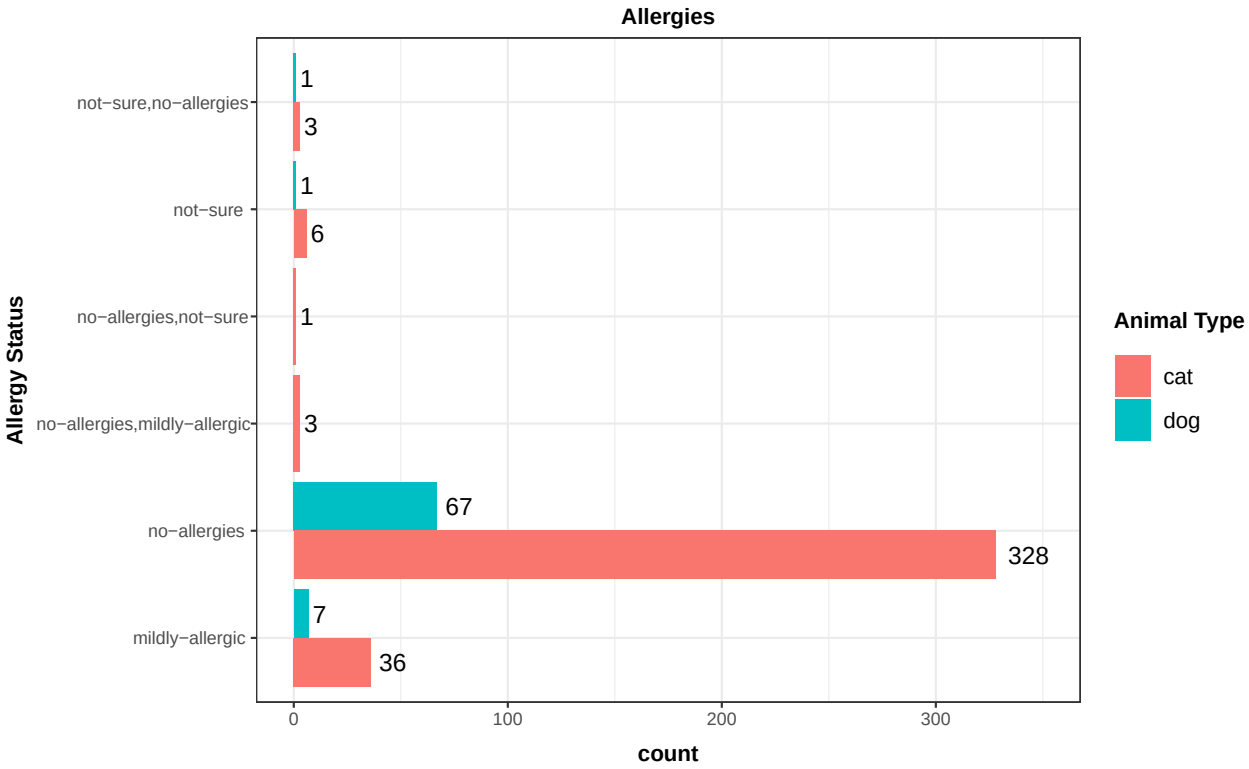
Interestingly, more number of applicants who were able to successfully adopt had less expereince in each of the types of experiences.



Not surprisingly, the highest number of succesful adoptions were associated with a home policy that allowed pets.



The main reason that people would return a pet in the future is if the pet sheds or if they move far away. Of those who said they would return the pet if it sheds, they either did not adopt or they adopted a cat.



Most of the people who adopted a pet had no allergies.

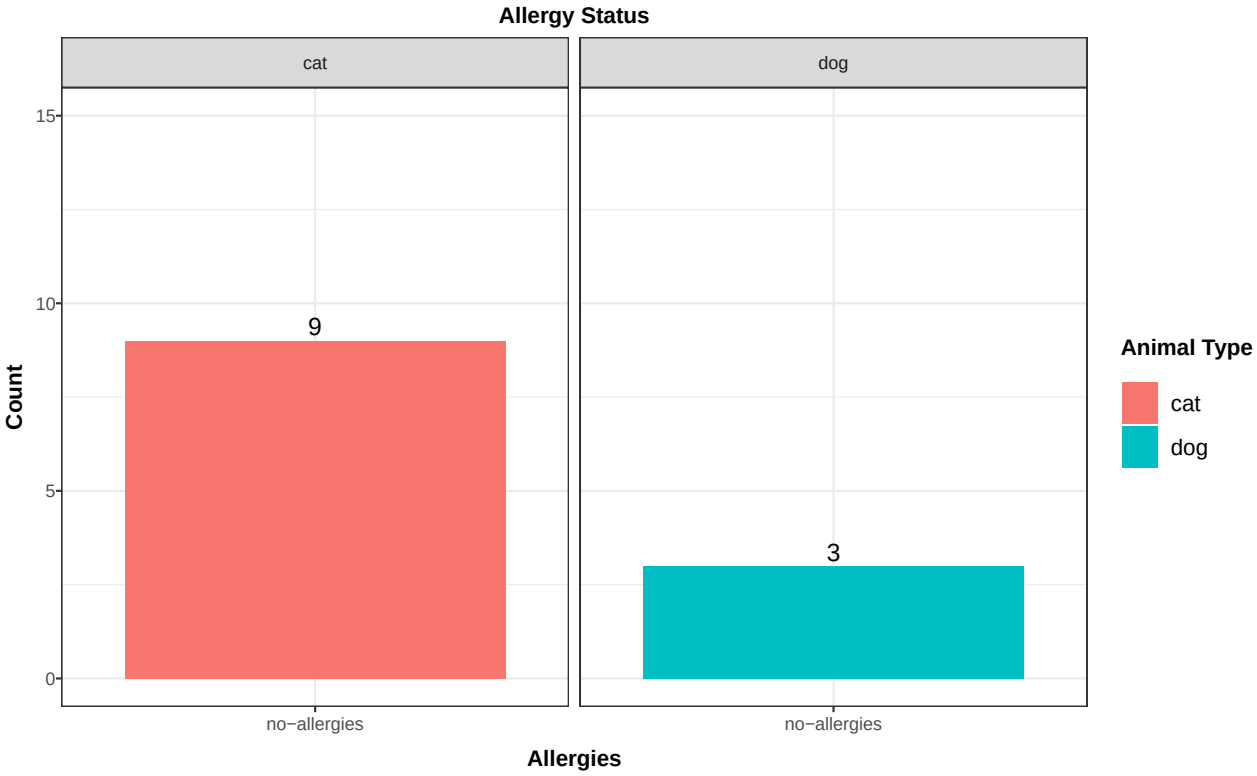
Analysis of Denied and Red Flagged Applications

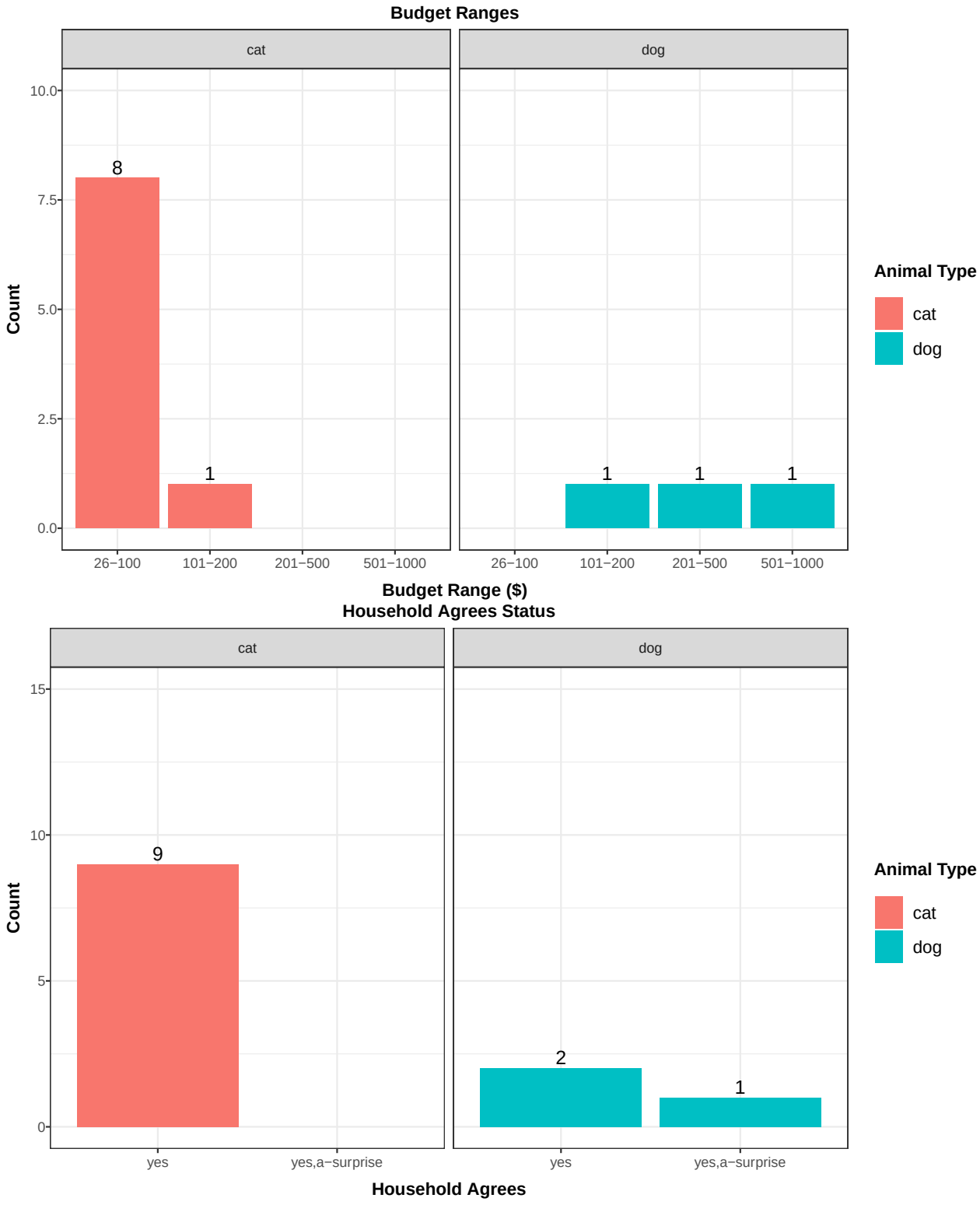
We further investigated the characteristics of applications that were denied or red flagged. There were 12 applications that were denied, 19 that were withdrawn, and 133 that were red flagged.

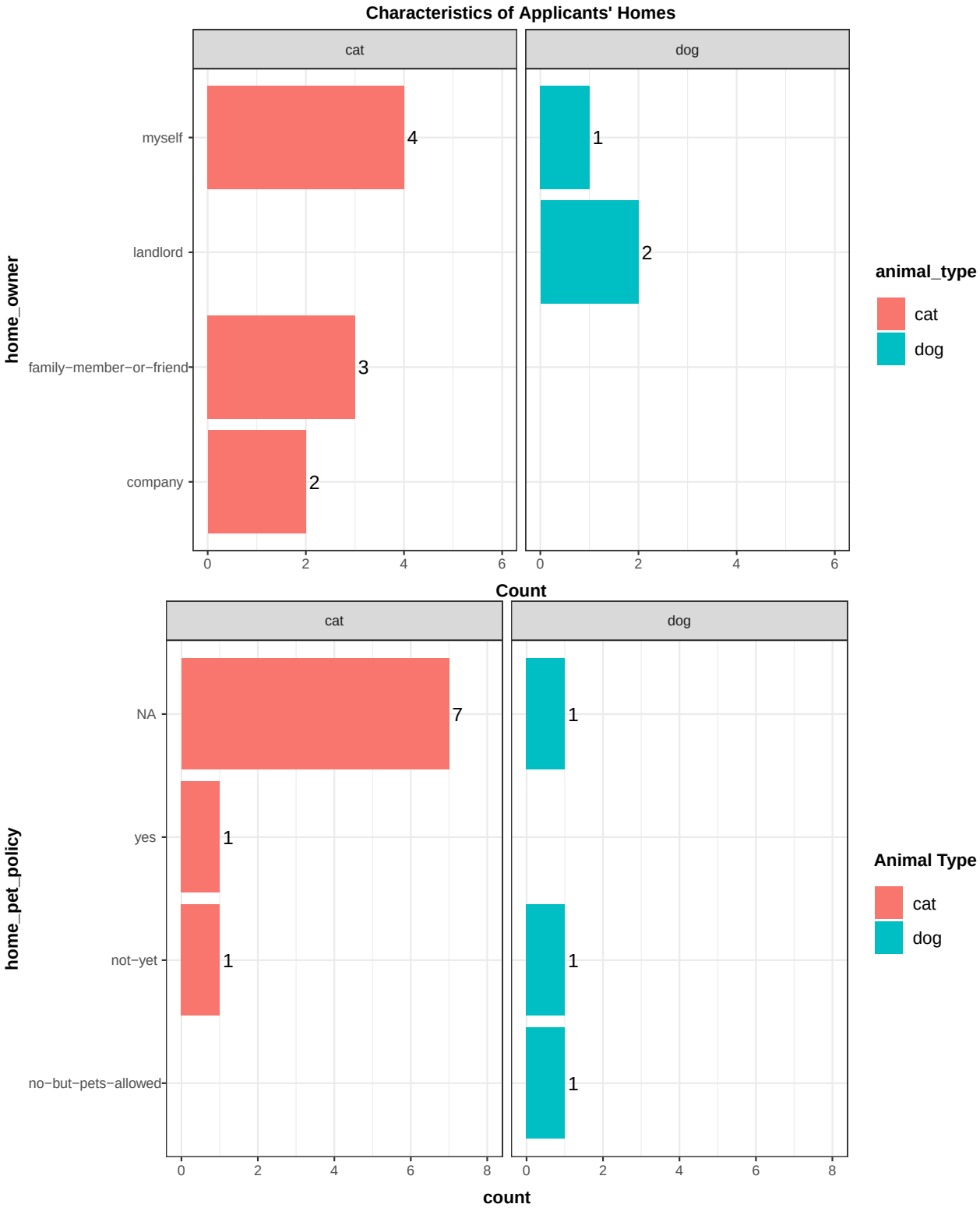
Denied Applications Below are visualizations that illustrate the applicants' characteristics (e.g. allergies, budget, home pet policy, etc.). We only have data for 12 denied applications so the analysis is limited. In the future when we have more data, we could compare the denied applications to the adopted ones.

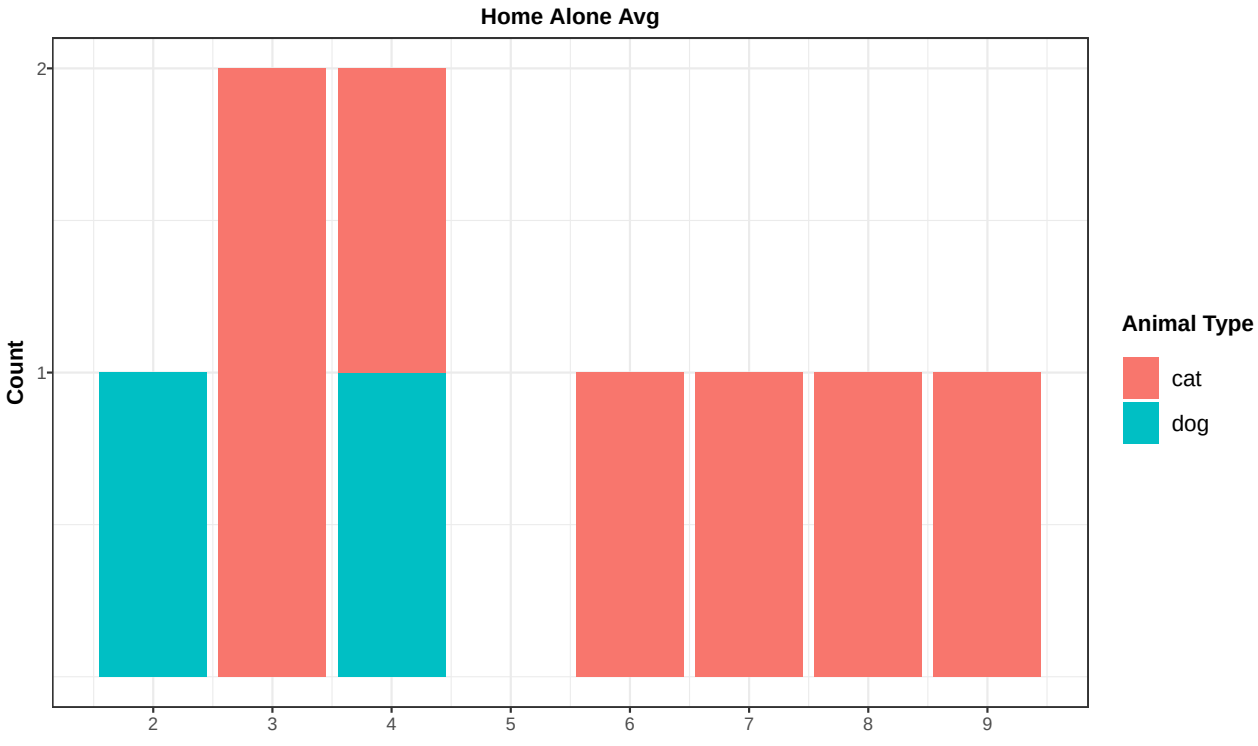
Key takeaways:

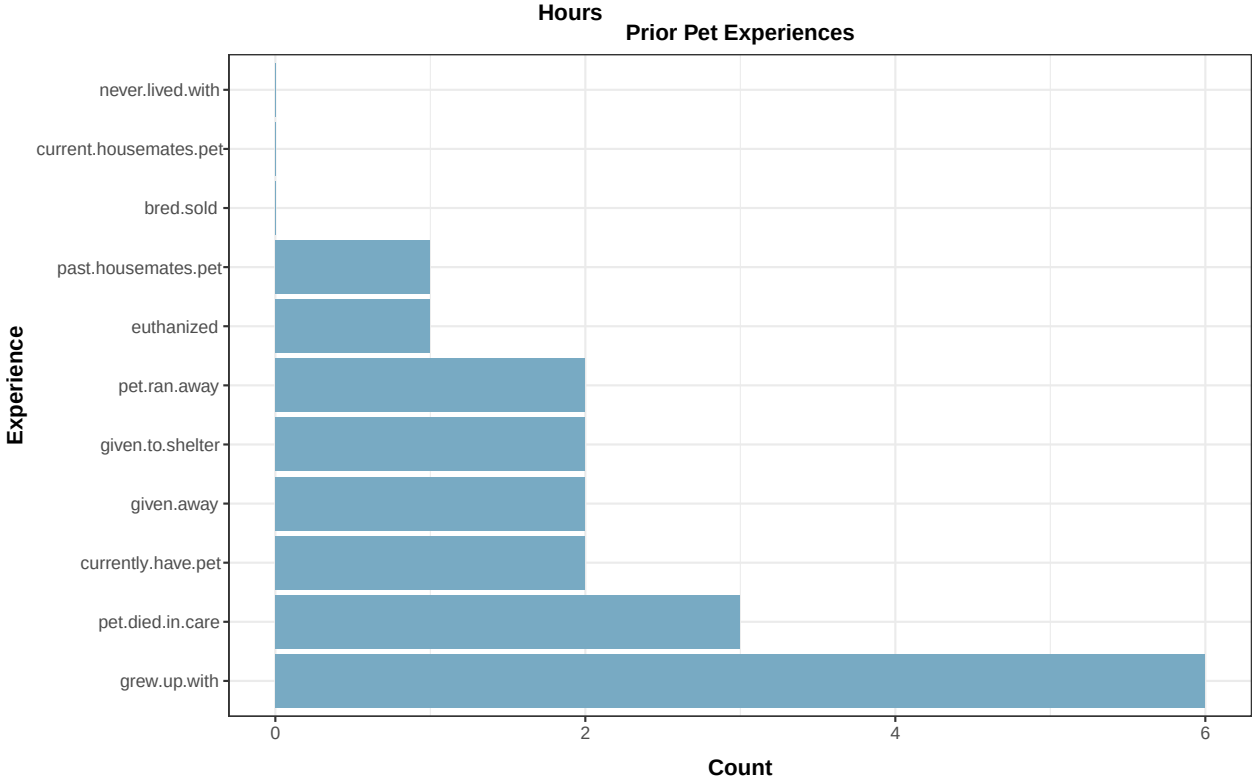
- No known allergies for the applicants
- Budget had no impact (same budget range for approved applications)
- All household members agreed to get a pet
- Majority of the applicants did not enter a home pet policy and not everyone is the home owner
- Many applicants had unfortunate incidents with prior pets (e.g. ran away, died in care)




Budget Ranges
cat
dog
10.0
7.5
5.0
2.5
0.0
Count
8
1
1
1
1
26−100
101−200
201−500
501−1000
Budget Range ($)
Household Agrees Status
15
10
5
0
9
2
1
yes
yes,a−surprise
Household Agrees
Animal Type
cat
dog


Characteristics of Applicants' Homes
cat
dog
myself
landlord
family−member−or−friend
company
home_owner
4
1
2
3
2
0
2
4
6
Count
animal_type
cat
dog
cat
dog
NA
yes
not−yet
no−but−pets−allowed
home_pet_policy
7
1
1
1
1
1
0
2
4
6
8
count
Animal Type
cat
dog



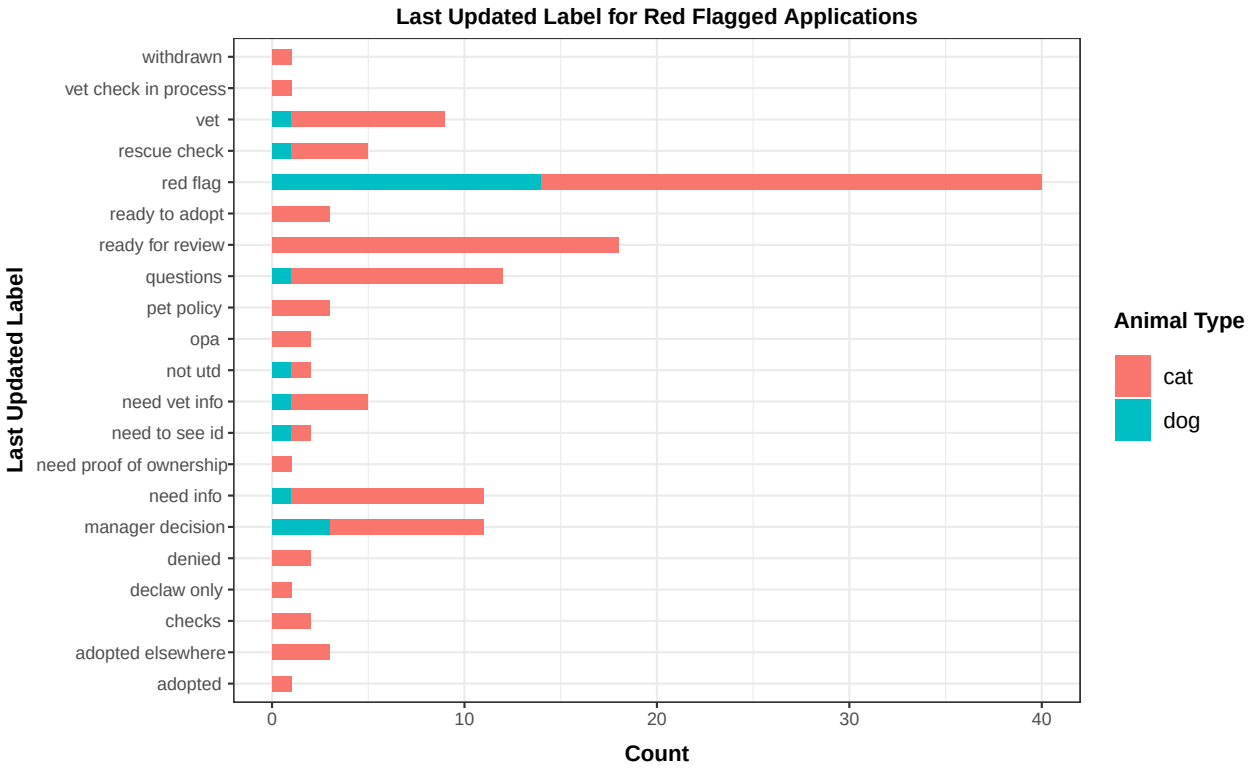


Red Flagged Applications

There were 133 applications that were red flagged.129 of the 133 have not yet resulted in an adoption or are still being procesed. Two of the applications that were flagged were denied but that does not mean that the rest are going to result in adoption. Since the data set for the applications is from the end of 2018, many of the applications are still in progress. We do not have the final status of all the applications so we cannot conclude what happened to the red flagged applications. As a further project, I think it would be interesting

to track the final status of the applications that were red flagged.

Below is a visualization that shows the last updated status for applications that were red flagged. After being flagged, the applications were sent to the manager to make a decision or the applicant was requested to provide more information (e.g. in many cases the applicant was required to provide more information about the vet).



Important Features for Prediction

Until now, we have separately analysed the different characteristics that affect adoption or decline. In an attempt to understand how the different features in the dataset could have had a combined effect on the adoption status, we ran a basic Random Forests model on the dataset. A Random Forest is basically a tree-based algorithm where a random subset of predictors (or features) are evaluated at each node and the observed data is split into two regions using one of the predictors and a threshold value for that predictor such that the error in predicting the adoption status is minimized. Starting from the top of the tree with one node, two new nodes are created with each split and the tree is grown recursively till there are only a few observations in each leaf node. Multiple trees are built similarly and the results are combined together to predict the adoption status for any given set of characteristics.

To successfully build a Random Forest, we further cleaned the data to take care of all the missing values. We used 1665 observations and 90 variables out of a total of 1684 observations and 251 variables.

The combined effect of different characteristics on the adoption status can be studied by considering one of the important outputs generated by the Random Forests, the subset of predictor values that are found to be most commonly used as a criteria for splitting the dataset into two smaller regions at each node. This subset of predictor values, referred to as Important Variables, are shown in the plot below. As seen in the list, we find that the top three characteristics are number of children in a home, the type of dog, and the date the application was submitted. Improved results or a different set of important characteristics can be obtained from better and more complete data.

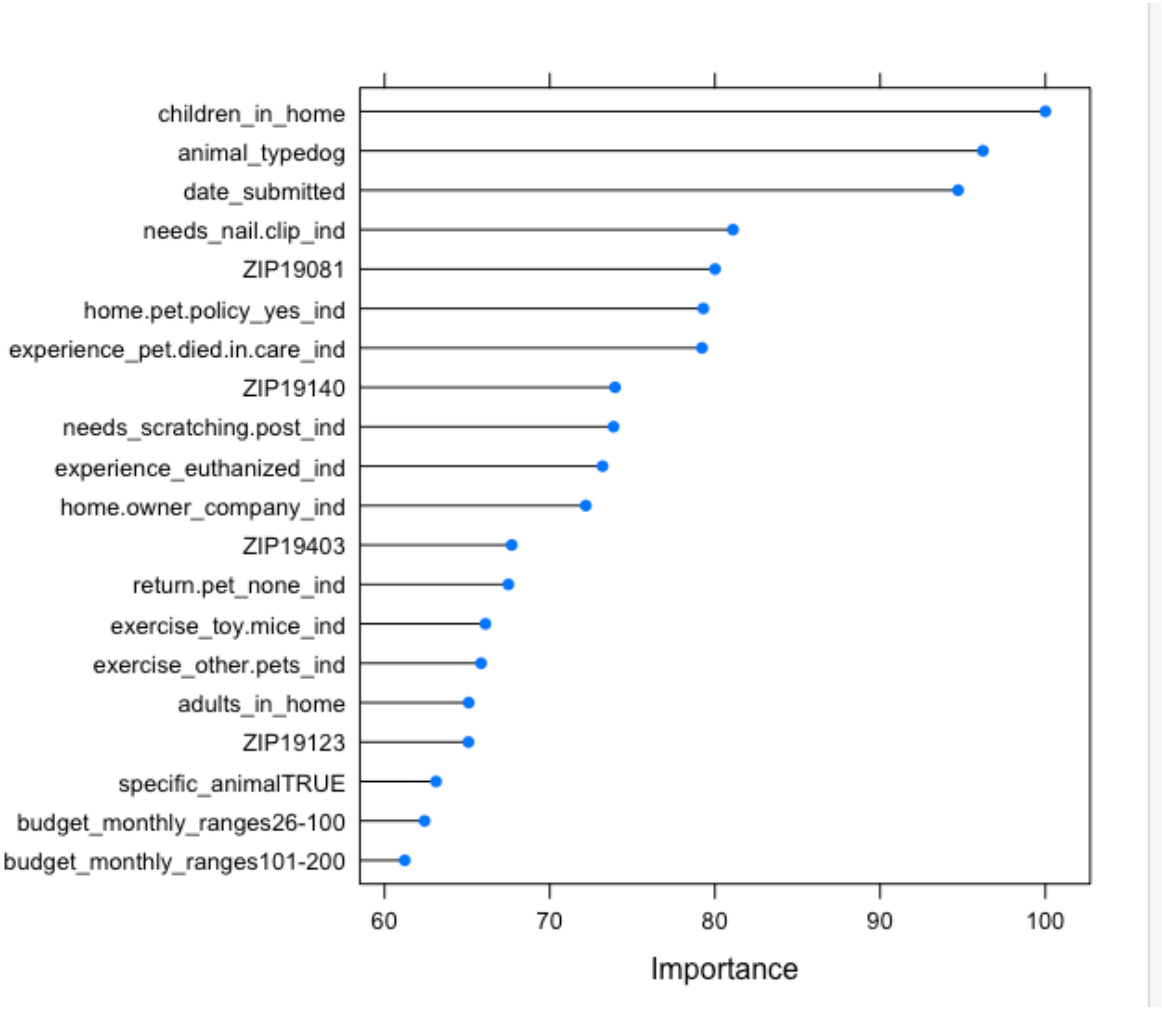


Figure 3: Important variables

3. Geographic factors

Contributors

Joy Payton, MS is the Supervisor of Data Education in the Department of Biomedical and Health Informatics at the Children's Hospital of Philadelphia. She leads the development and implementation of education and outreach programs to help CHOP scientists become data-savvy and make the best, most informed use of the tools they have available.

Karla Fettich, PhD is Head of Algorithm Development at Orchestrall, Inc. She leads efforts to develop data analytics solutions, predictive models and optimization approaches to create sustainable changes that improve operations and outcomes in long term care facilities.

Goals

These analyses examined the data in relation to geographic and population parameters, with two main objectives:

1) identify an initial set of variables that may be informative for application processing
2) provide a basis for discussion around the usefulness of geographical data analysis for PAWS at a broader level

Datasets

The following datasets were used:

1. Online applications for both cats and dogs In addition to the data collected via the online forms, applicants' addresses were extracted and associated with their respective census tracts. Census tracts are areas roughly equivalent to a neighborhood established by the Bureau of Census for analyzing populations, and generally have a population size between 1,200 and 8,000 people, with an optimum size of 4,000 people. Prior to making the PAWS data available, individual applicants' names, addresses and other identifiable data were removed from the dataset, keeping only census tract data and ZIP codes.
2. Trello cards and actions
3. Census data from the 2017 five-year American Community Survey via the American Fact Finder for the following areas:

- Economic characteristics
- Education characteristics
- Median rent
- Computer and networking characteristics

Results

Economic Considerations in Processing Applications

On average, dog applicants live in areas where the median income is higher compared to cat applicants (around $60,000/year for dog applicants vs. $54,000/year for cat applicants) and where the percent of households living under the poverty level is lower (18% for dog applicants vs. 22% for cat applicants). This suggests that dog applicants are from slightly wealthier neighborhoods. We further observed that dog applicants have more range between lower middle class and upper middle class, while cat applicants tend to skew more toward lower incomes. This finding aligns with the pet care cost estimates provided by the ASPCA which

suggest that the first year total costs of owning a dog ($1,471 - $1,779) exceed those of owning a cat ($1,174) - although it is unclear how recent these estimates are.

Using the "complete" status of a trello card at the time when the data were pulled, we did not observe a neighborhood wealth difference between completed and non-completed applications. While a "complete" status is fairly vague (it does not indicate the outcome of an application), and several trello cards may have been incomplete due to them being fairly recent, the data do not indicate an economic bias when processing applications.

We further looked into some of the outcomes of application processing, specifically *red flags* and *denied* applications. Applications from neighborhoods with a lower household median income (under $50,000/year) are more likely to be red flagged and denied, compared to those with a higher household median income (over $50,000/year). Additionally, red flagged **cat** applicants have a lower estimated monthly budget than their non-red-flagged counterparts ($176 vs. $224). For **dogs**, a similar trend was observed, but it did not reach the statistical significance threshold ($212 vs. $277). This pattern also holds when it comes to emergency budgets: red flagged applicants have a lower estimated emergency budget than their non-red-flagged counterparts ($947 vs. $1,446 for **cats** and $735 vs. $1,848 for **dogs**). While we found that living in a lower income neighborhood does impact the estimated emergency budget at a statistically significant level, it only accounts for about 7% of the observed pattern. This indicates that there are additional factors that may play a role in how much money an applicant is able to set aside on a regular basis for pet care.

Efficiency Analysis in Philadelphia County

We also looked at applications that were processed within an efficient timeframe (defined here as 10 days), vs those that did not. An application was considered efficient if it was given a decision label ("denied", "do not follow up", "adopted", "adoption follow up", "approved", "ready to adopt", "ready for review", "reviewed with handouts only", "approved with limitation", "dog meet", "returned", "adopted elsewhere") and the last trello checklist item was checked off 10 days or less from the date of application submission.

Dogs

We found that in neighborhoods with a higher percentage of people who have a cell data plan and no other type of internet subscription, there was also a trend for a lower proportion of efficient applications, this effect being more pronounced in north and northeast Philadephia. There could be many reasons for this: applicants who live in areas where many people do not have easy access to the internet may not be as familiar with filling out an online application (which represents the current application dataset); they may also not be able to easily find the information they need (since not all websites are mobile friendly); or they may be filling out the application form on a mobile device, which might be too long/detailed to adequately complete on a small screen.

Additionally, in neighborhoods with a higher percentage of the population 25 to 34 year old enrolled in school, we also observed a significantly higher proprotion of efficient applications. It is unclear what the reasons behind this might be, but possible options include the applicants' level of comfort with online applications, access to information, or other factors that are more specific to the life circumstances of individuals enrolled in school. This effect was less pronounced in northeast Philly.

Cats

Interestingly, for cats we found that in neighborhoods where a higher percentage of the population is children in grades 5-8, the proportion of efficient applications was lower, this effect being more pronounced in the north and northeast. While we do not know the reasons for this effect, it may be worth noting that ownership of and interest in pets tend to peak in middle childhood (i.e., 8–12 years). It may be that this effect influences the decision to submit an application, but that other barriers interfere with the application's timely processing (e.g. incomplete information, lack of responsiveness to provide additional information, change of mind).

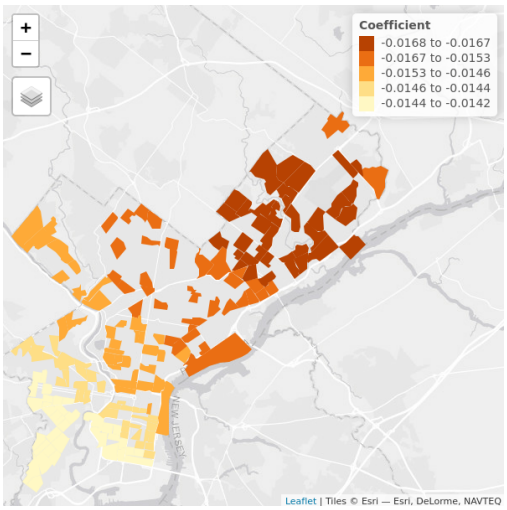


Figure 4: Cell data plan only coefficients

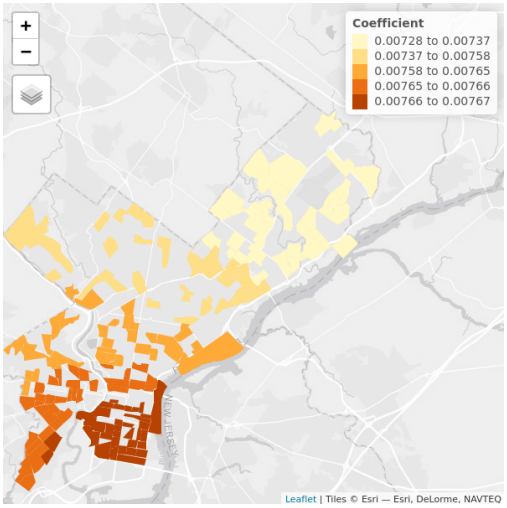


Figure 5: Population 25-34 enrolled in school coefficients

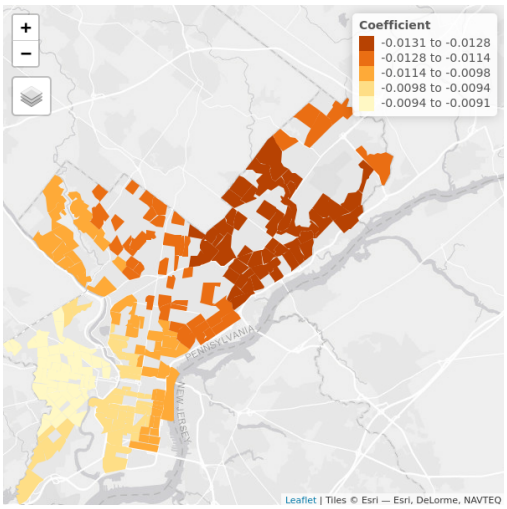


Figure 6: Children grades 5-8 coefficients

Conclusions and Next Steps

1. PAWS could develop a "smart" online application, that automatically educates the applicant on the cost of pet ownership when the budget is too low.

Since red flagged and denied applications still require processing by PAWS staff, and possibly even more intense processing than approved applications, it may be worth automatically screening and educating applicants who may have unrealistic budgeting expectations. Thus, perhaps a pop-up chart could appear when the budget is too low, *while* the applicant fills out the form. If the applicant proceeds to submit the application with a too-low budget, this application could be automatically labeled a red flag and sent for processing to a more experienced staff for further processing.

2. PAWS could provide applicants with a detailed breakdown of costs for a new pet, and have an adoption counselor go through the itemized list with the applicant to identify how each item could be covered.

Taking for instance the pet care cost estimates provided by the ASPCA, PAWS could identify which categories might be most difficult for an applicant to cover. Then, PAWS could provide a set of options (e.g. list of lower cost vets, cheaper options for enrichment using household items, list of affordable dog trainers) that might make the costs more manageable for those who are on a tighter budget.

3. PAWS could promote sharing or pooling of resources among its adopters.

Many pets have preferences when it comes to food, treats and toys, and it takes a while for a new adopter to learn them. This can result in a lot of wasted money. PAWS could facilitate and promote sharing of these resources (including any other accessories, or even transport help), at the adopters' own risk, via an online community.

4. PAWS could assess the user-friendliness of its online application form on different platforms.

While the PAWS website might be mbile-friendly, PAWS could further assess whether the application form itself is represented in the most efficient way on a mobile device. To do this, information would first need to

be collected on the number of applicants who submit the application from a mobile device, as a revamping of the mobile interface for the application form may only be necessary if application quality is dependent on the device from which the application was submitted. An additional indicator of user friendliness could be the amount of time applicants spend on an application. PAWS could consider a 'smart' approach in sequencing and presenting questions so that the process is relatively speedy for the applicant, while also ensuring quality data.

5. PAWS could consider creating programs that are aimed at families with middle-schoolers.

Given that there is a spike in children's interest in animals in middle school, PAWS could consider some ways to increase involvement of children in the animal care process, either by creating kid-friendly volunteer opportunities, kid-friendly community groups among adopters, or even informational events where people who are interested in adopting can ask questions and discuss experiences with adopters and PAWS representatives.

4. Social media factors

- @phillypaws tweets were analyzed to understand any patterns in twitter activity and whether that could be linked to application or pet information
- No strong trends or observations were gained, but it was fun to look at the data
- We did not see a strong association between twitter activity and applications
- We identified the most commonly tweeted words such as “home”, “adoption”, and “meet”
- We found that tweets with photos were more often favorited and retweeted

Contributors

Alice M Walsh, PhD is a computational biologist in the pharmaceutical industry. She enjoys analyzing patient data and trying to make informed decisions using data. She also enjoys walking and training her dog, Pebbles.

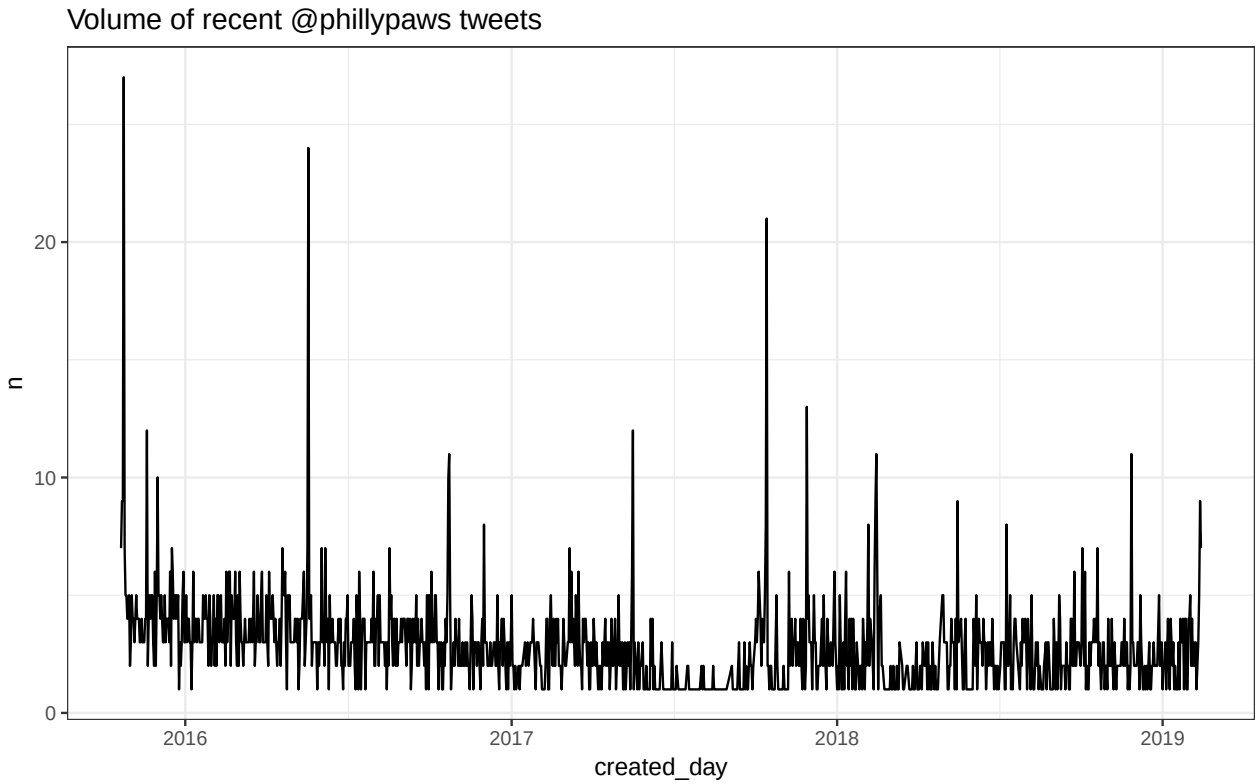
Problem definition and dataset

We examined the data from the PAWS twitter account, @phillypaws. We used the twitter API to download the most recent 3200 tweets, which included all tweets from 2018. Quotes and retweets were not excluded from the dataset.

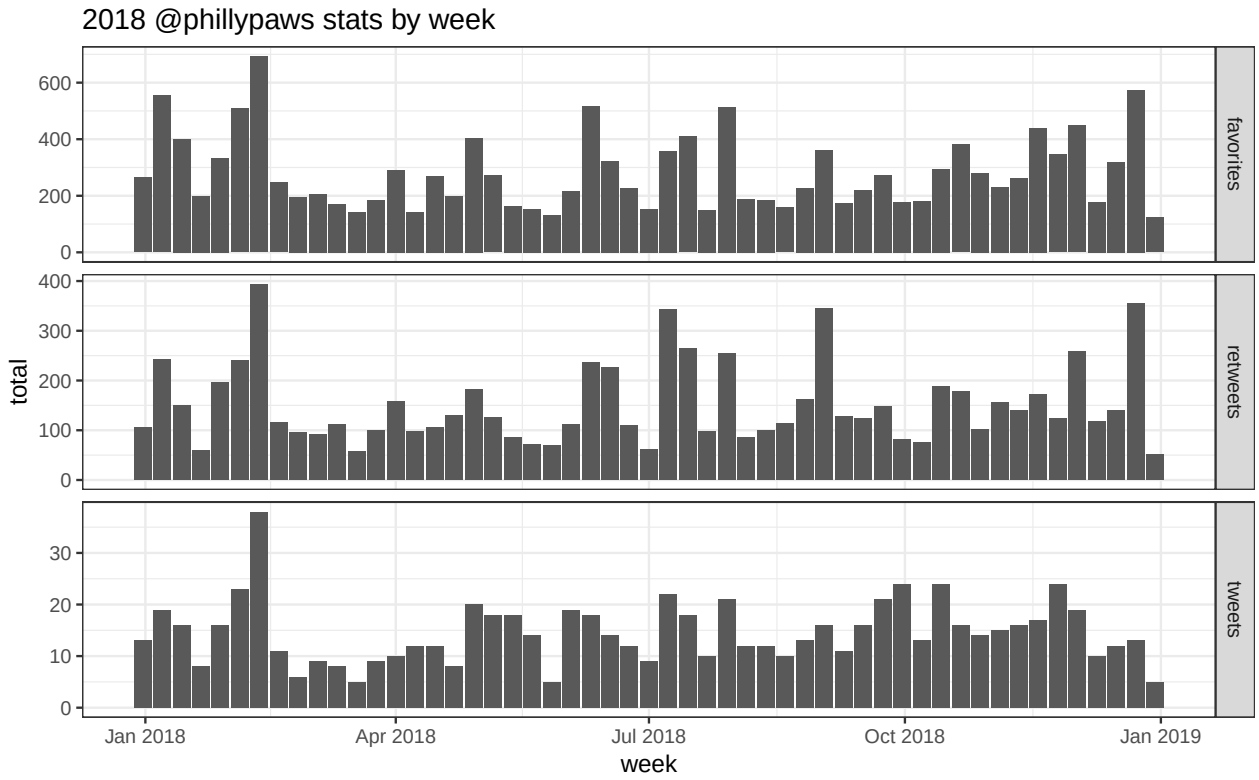
Results

```
# Load in data - previously pulled with rtweet package
tweets <- readRDS(here::here('/Analyses/4_Other/tweets_13FEB2019.Rds'))
# Let's focus on 2018 to match other datasets
tweets_18 <- filter(tweets, created_at < as.Date("2019-01-01"), created_at > as.Date("2018-01-01"))
```

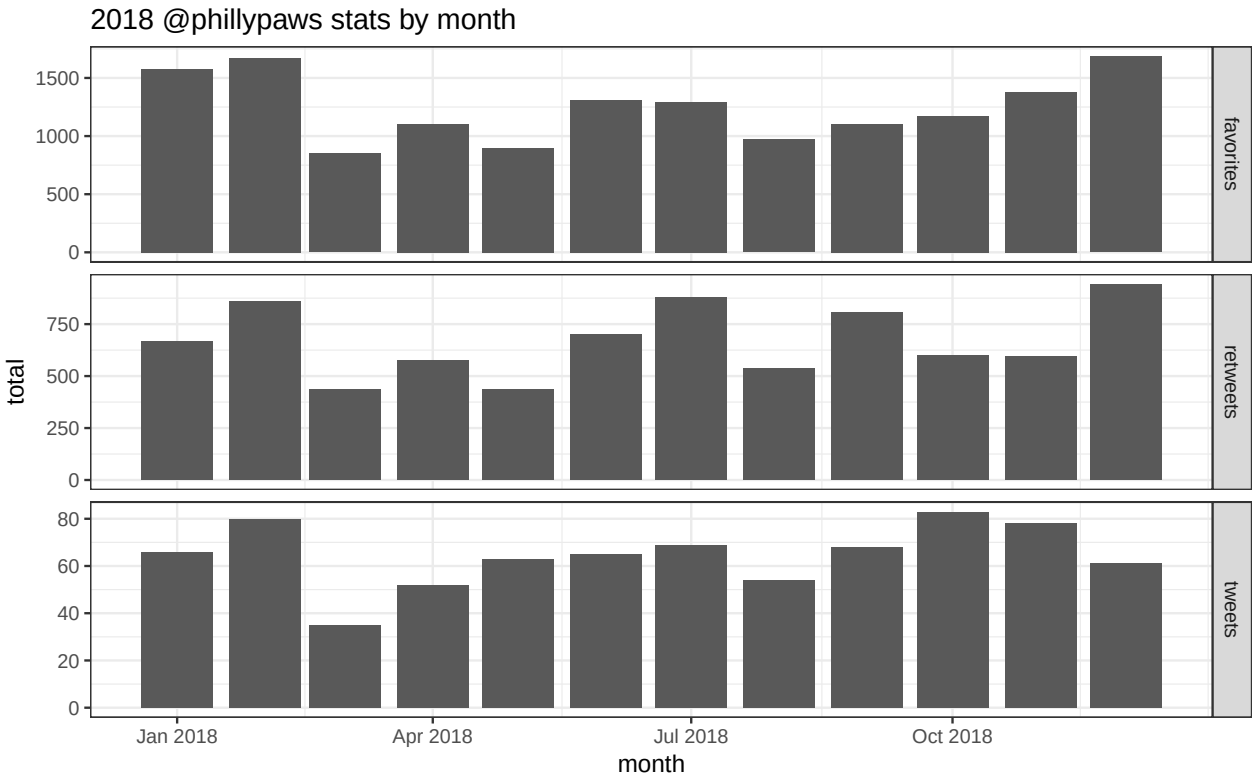
```
tweets %>%
  mutate(created_day = lubridate::floor_date(created_at, unit = "day")) %>%
  count(created_day) %>%
  ggplot(aes(x=created_day, y=n)) +
  geom_line() +
  theme_bw() +
  ggtitle("Volume of recent @phillypaws tweets")
```



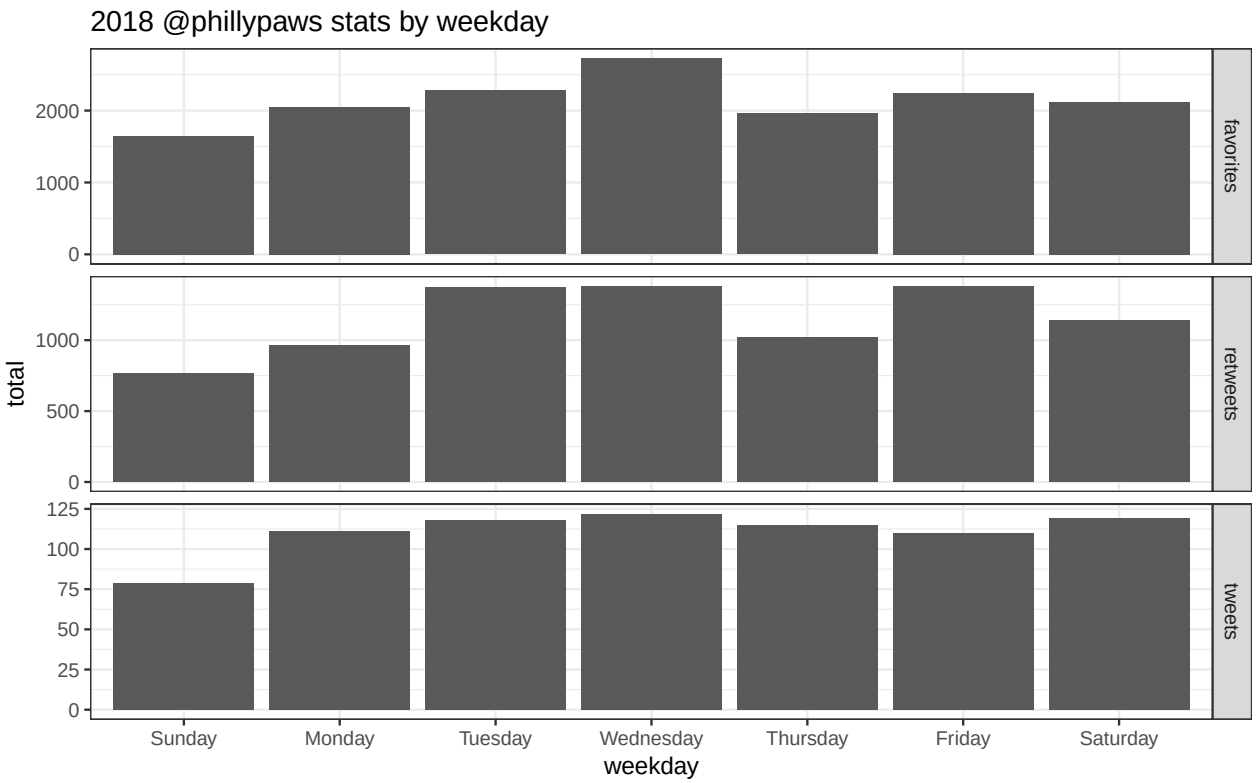

```
# Let's count tweets by day-of-the-week, week, month, and look for trends
# Plot by week:
tweets_18 %>%
  mutate(week = lubridate::floor_date(created_at, unit = "week")) %>%
  group_by(week) %>%
  summarise(favorites = sum(favorite_count),
            tweets = n(),
            retweets = sum(retweet_count)) %>%
  gather(metric, total, favorites:retweets) %>%
  ggplot(aes(x=week, y=total)) +
  geom_col() +
  facet_grid(metric~., scales = "free") +
  theme_bw() +
  ggtitle("2018 @phillypaws stats by week")
```




```
# Plot by month:
tweets_18 %>%
  mutate(month = lubridate::floor_date(created_at, unit = "month")) %>%
  group_by(month) %>%
  summarise(favorites = sum(favorite_count),
            tweets = n(),
            retweets = sum(retweet_count)) %>%
  gather(metric, total, favorites:retweets) %>%
  ggplot(aes(x=month, y=total)) +
  geom_col() +
  facet_grid(metric~., scales = "free") +
  theme_bw() +
  ggtitle("2018 @phillypaws stats by month")
```

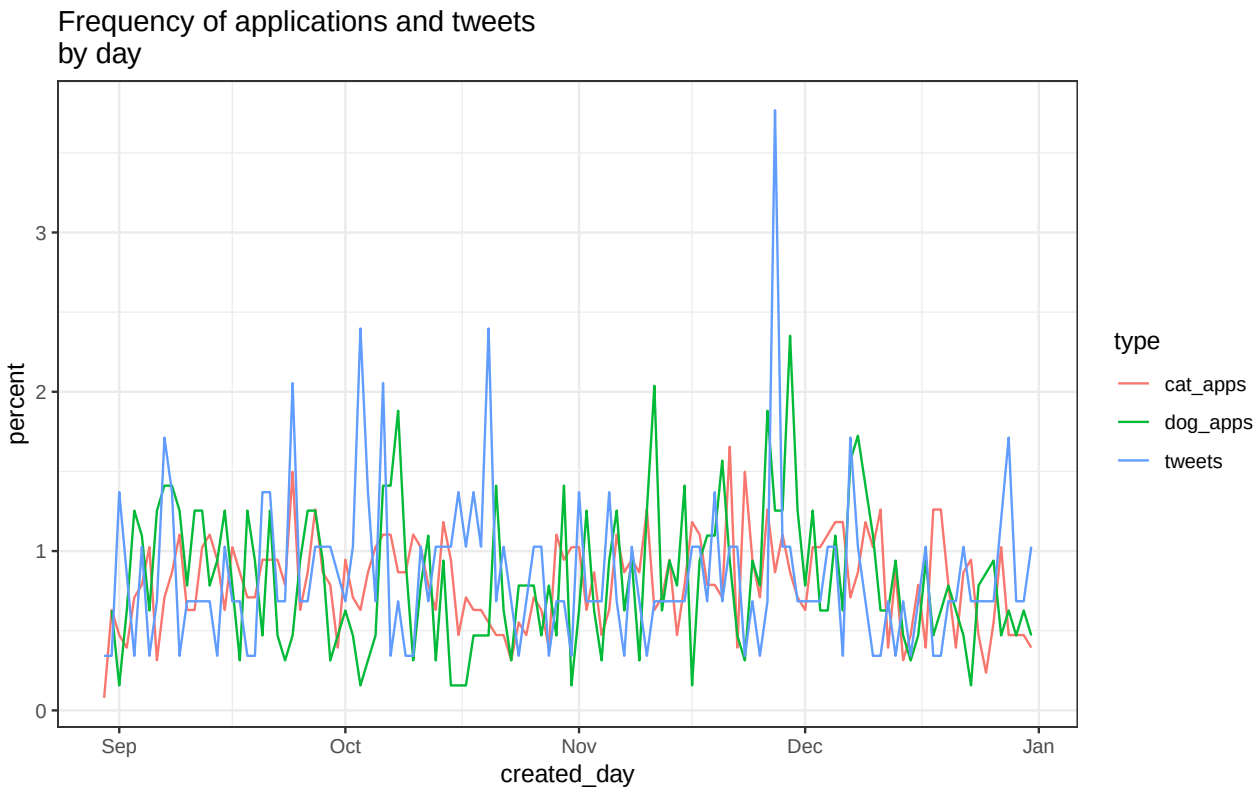



```
# Plot by day-of-the-week:
tweets_18 %>%
  mutate(weekday = weekdays(created_at)) %>%
  mutate(weekday = factor(weekday,
                          levels = c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","S
  group_by(weekday) %>%
  summarise(favorites = sum(favorite_count),
            tweets = n(),
            retweets = sum(retweet_count)) %>%
  gather(metric, total, favorites:retweets) %>%
  ggplot(aes(x=weekday, y=total)) +
  geom_col() +
  facet_grid(metric~., scales = "free") +
  theme_bw() +
  ggtitle("2018 @phillypaws stats by weekday")
```

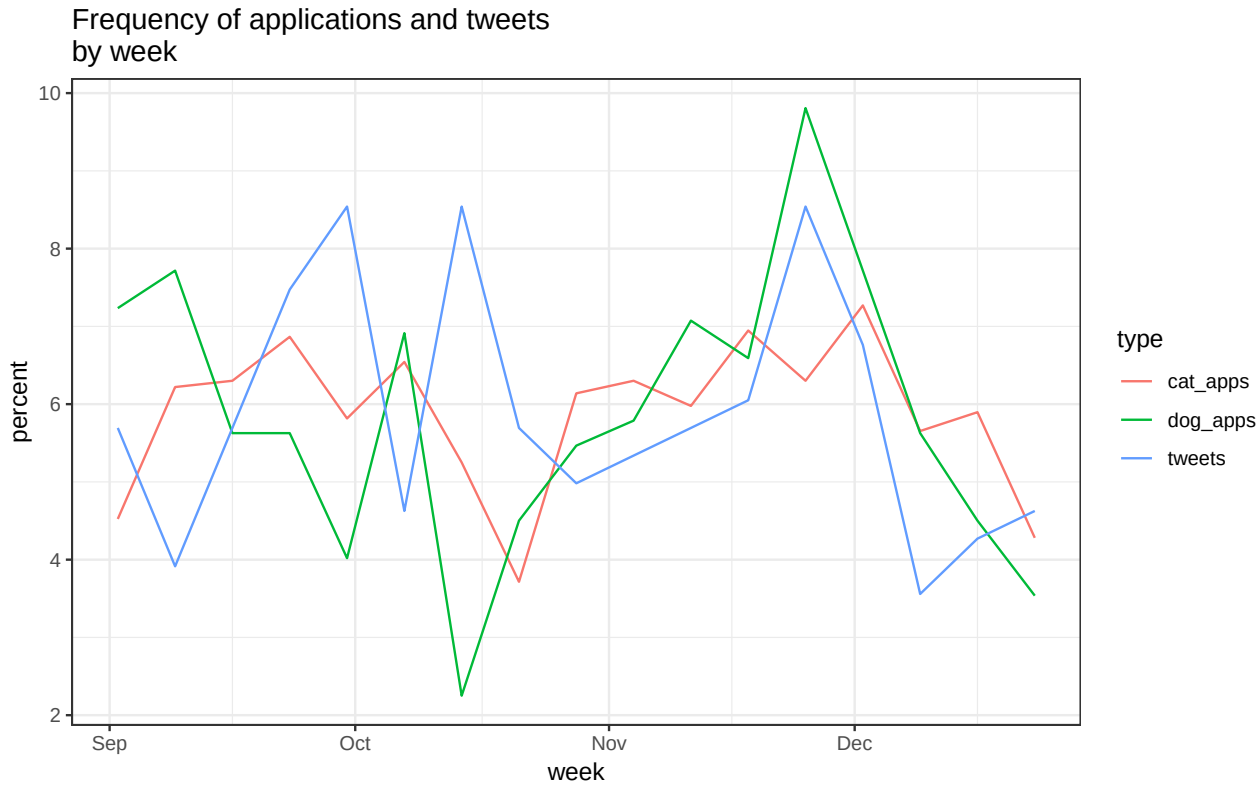



```
# Compare twitter activity to application activity?
cat_apps <- read.csv(here::here('/Data/cat_apps.csv'),
                     na.strings = c(" ","","na","NA"), stringsAsFactors = F)%>%
  janitor::clean_names() %>%
  mutate(date_submitted = as.Date(date_submitted, "%m/%d/%Y"))

dog_apps <- read.csv((here::here('/Data/dog_apps.csv')),
                     na.strings = c(" ","","na","NA"), stringsAsFactors = F)%>%
  janitor::clean_names() %>%
  mutate(date_submitted = as.Date(date_submitted, "%m/%d/%Y"))

compare_cat_apps <- cat_apps %>%
  count(date_submitted) %>%
  mutate(type = "cat_apps") %>%
  rename(created_day = date_submitted)

compare_dog_apps <- dog_apps %>%
  count(date_submitted) %>%
  mutate(type = "dog_apps") %>%
  rename(created_day = date_submitted)

compare_tweets <- tweets_18 %>%
  mutate(created_day = lubridate::floor_date(created_at, unit = "day")) %>%
  filter(created_day >= min(compare_cat_apps$created_day)) %>%
  count(created_day) %>%
  mutate(type = "tweets")

compare_days  <- rbind(compare_tweets, compare_cat_apps, compare_dog_apps)
```

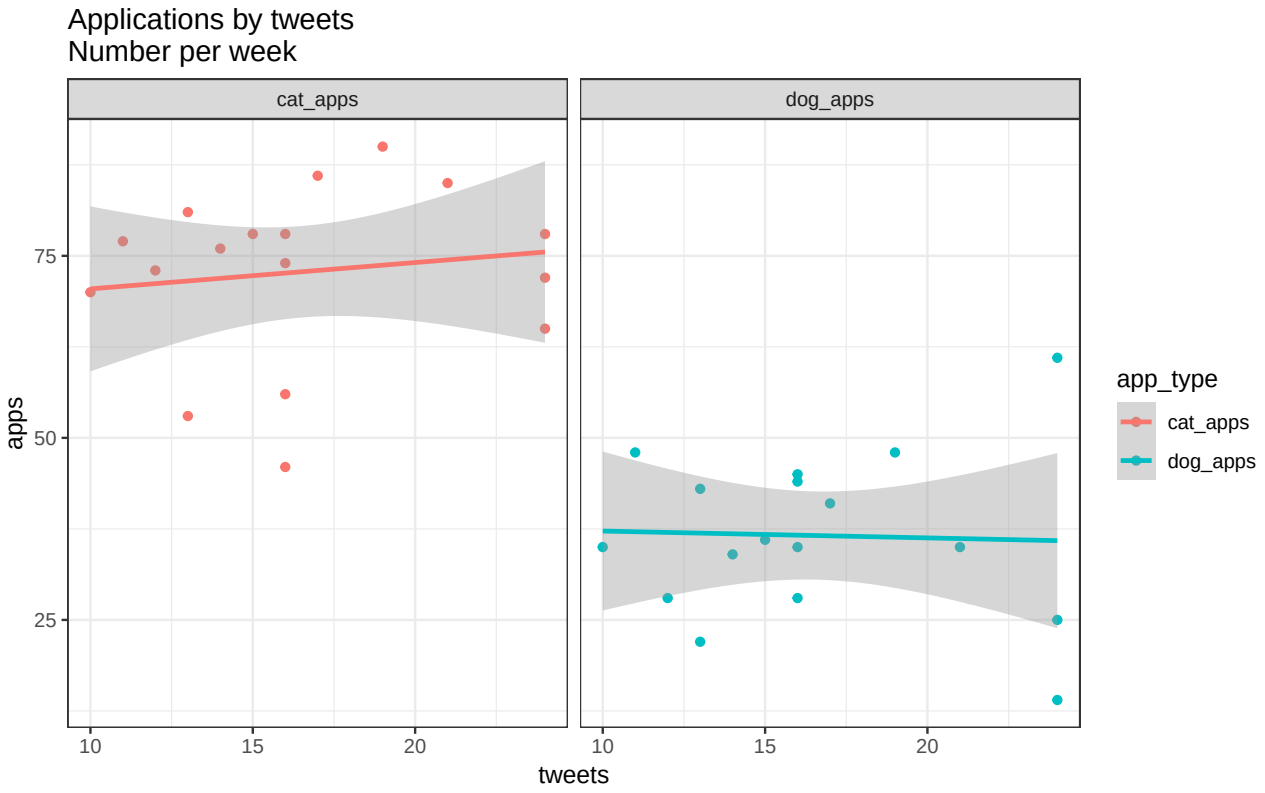
```r
# Line by day
compare_days %>%
  group_by(type) %>%
  mutate(percent = n/sum(n) * 100) %>%
  ggplot(aes(x=created_day, y=percent, color=type)) +
  geom_line() +
  theme_bw()+
  ggtitle("Frequency of applications and tweets \nby day")
```



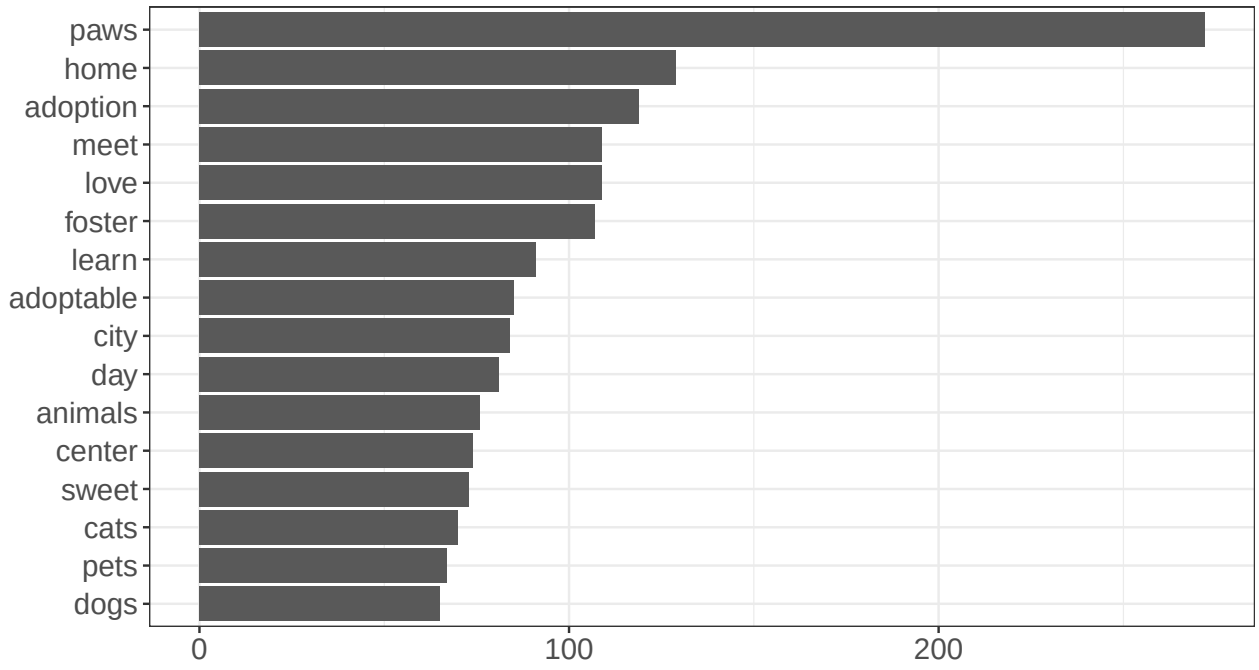

```r
# Line by week
compare_days %>%
  # remove the short week - 2018-08-26 and 2018-12-30
  filter(created_day > as.Date("2018-09-02"), created_day < as.Date("2018-12-30")) %>%
  mutate(week = lubridate::floor_date(created_day, unit = "week")) %>%
  group_by(type, week) %>%
  summarise(total = sum(n)) %>%
  mutate(percent = total/sum(total) * 100) %>%
  ggplot(aes(x=week, y = percent)) +
  # geom_bar(aes(fill = type), position="dodge", stat = "identity") +
  geom_line(aes(color = type))+
  theme_bw() +
  ggtitle("Frequency of applications and tweets \nby week")
```



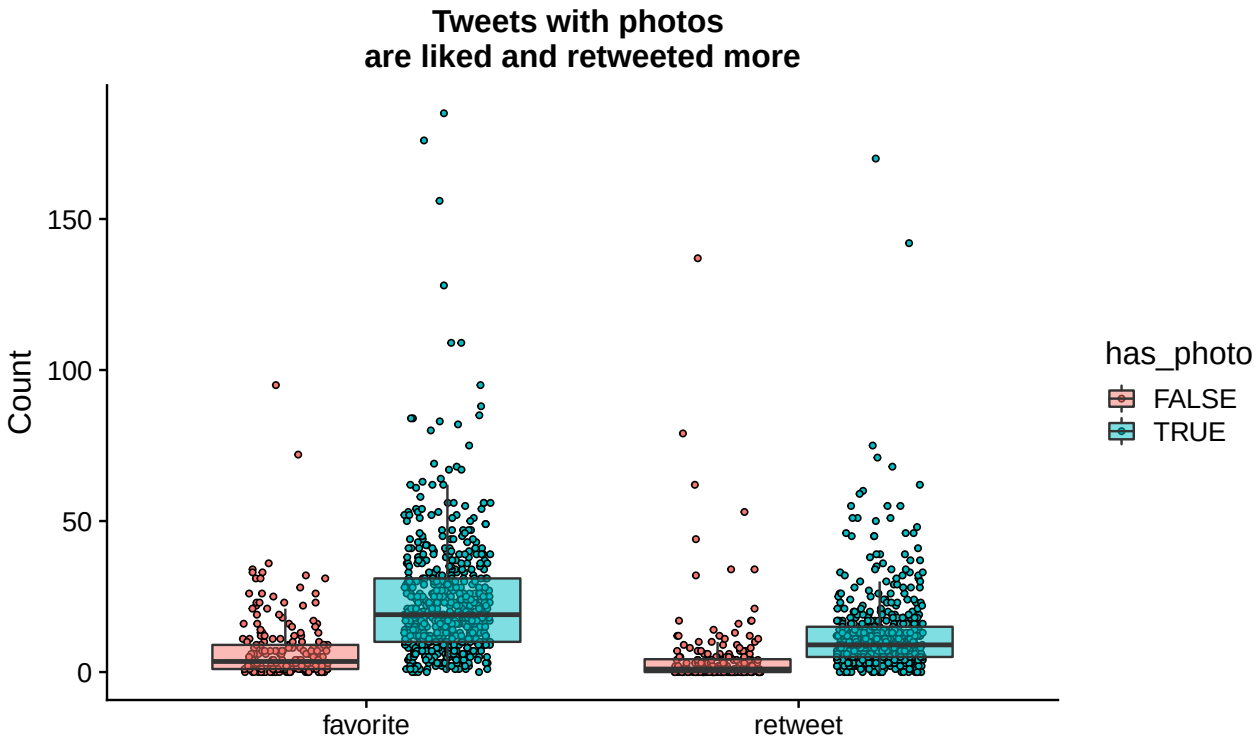

```
# Scatter plot grouped by week
compare_days %>%
  # remove the short week - 2018-08-26 and 2018-12-30
  filter(created_day > as.Date("2018-09-02"), created_day < as.Date("2018-12-30")) %>%
  mutate(week = lubridate::floor_date(created_day, unit = "week")) %>%
  group_by(type, week) %>%
  summarise(total = sum(n)) %>%
  ungroup() %>%
  tidyr::spread(key = type, value = total) %>%
  tidyr::gather(app_type, apps, cat_apps:dog_apps) %>%
  ggplot(aes(x=tweets, y = apps, color=app_type)) +
  geom_point() +
  geom_smooth(method = "lm") +
  facet_wrap(~app_type) +
  theme_bw() +
  ggtitle("Applications by tweets\nNumber per week")
```




```
# What are the top words in all the tweets?
# Wrote a function to plot for a given dataset
# thanks to https://www.williamrchase.com/post/is-hadley-wickham-a-cat-or-dog-person-a-twitter-tidytext
source(here::here('/Analyses/4_Other/alice_plot_top_words.R'))

# plot_top_words(tweets)
plot_top_words(tweets_18) + ggtitle("@phillypaws 2018 tweets\nWord frequency") + theme_bw() + xlab("") +
```

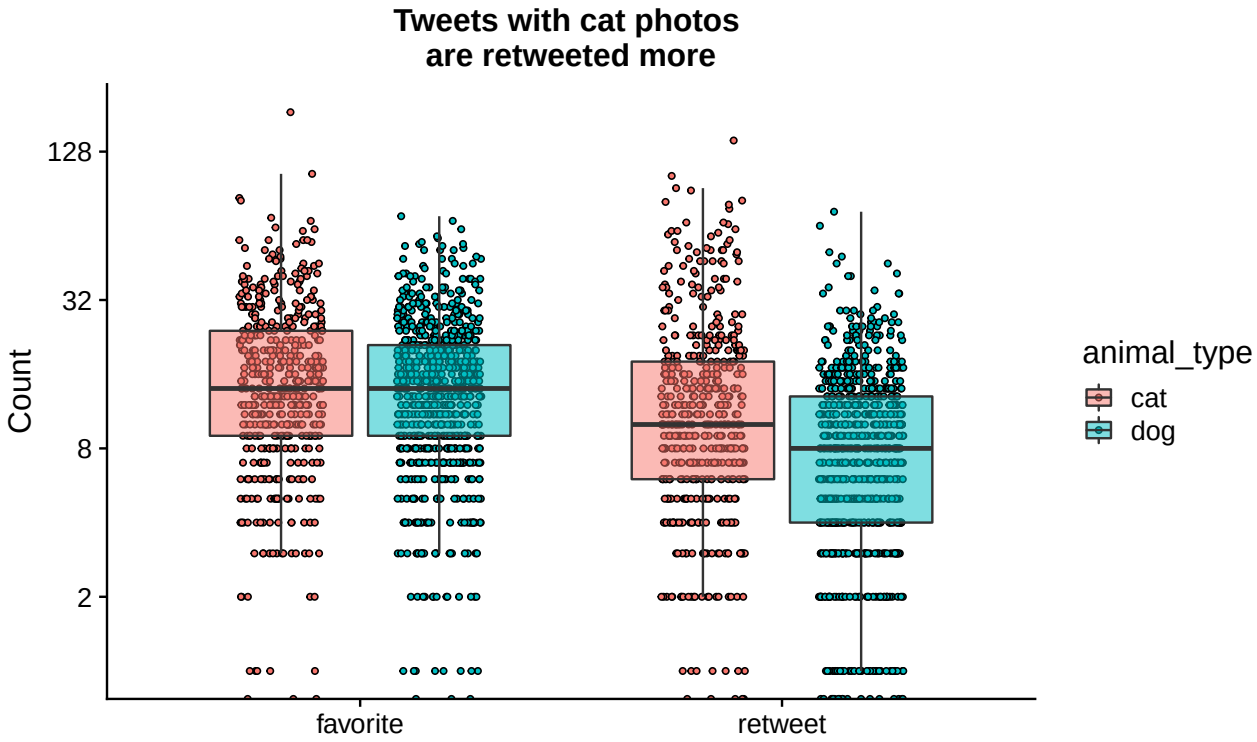



```
tweets_18 %>%
  mutate(has_photo = !is.na(media_type)) %>%
  gather(metric, count, favorite_count:retweet_count) %>%
  mutate(metric = gsub("_count","",metric)) %>%
  ggplot(aes(x=metric, y=count, fill = has_photo)) +
  geom_point(pch=21, size=1,
             position = position_jitterdodge(jitter.width = 0.2, jitter.height = 0)) +
  geom_boxplot(alpha = 0.5, outlier.shape = NA) +
  ylab("Count") + xlab("") +
  ggtitle("Tweets with photos \nare liked and retweeted more")
```

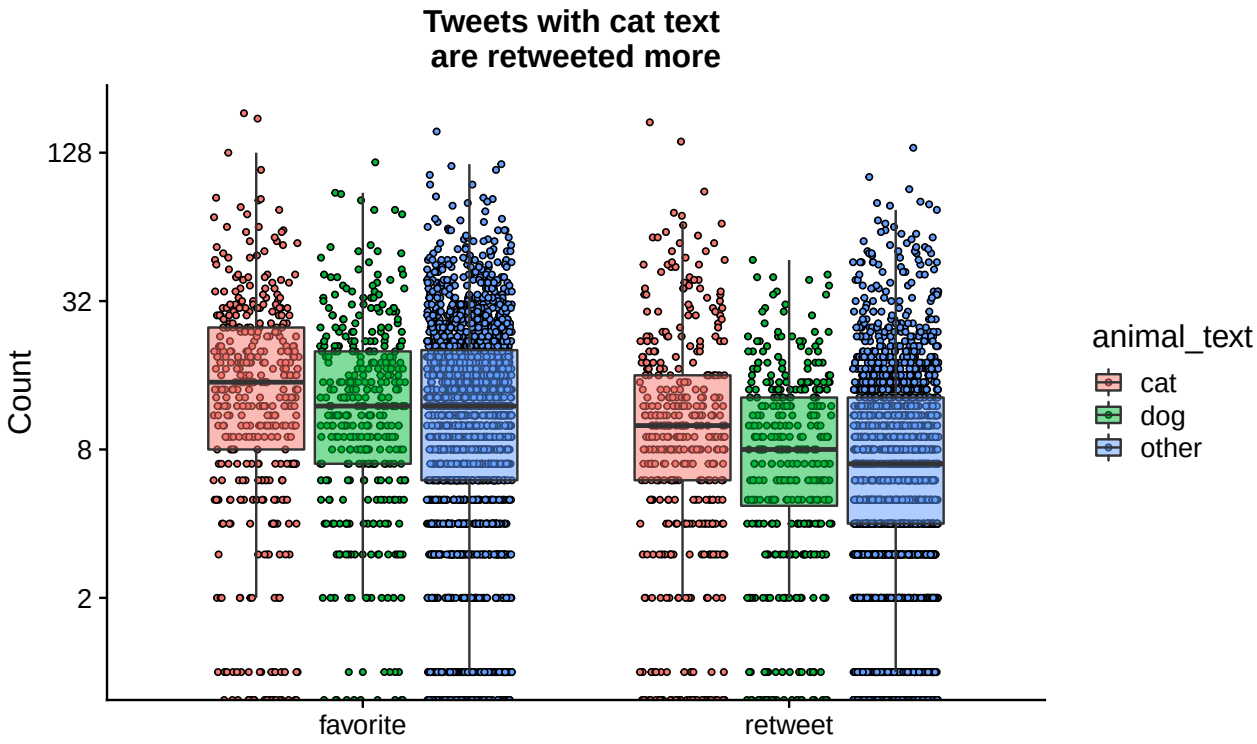



```
# Merge in cat/dog predictions
pred_img <- readRDS(here::here('/Analyses/4_Other/alice_predict.Rds'))
pred_img <- pred_img %>%
  mutate(status_id = gsub(".jpg","",gsub("predict/","",filenames)))

tweets_pred <- tweets %>%
  left_join(pred_img, by = "status_id") %>%
  mutate(has_photo = !is.na(media_type)) %>%
  # filter(has_photo == T) %>%
  mutate(animal_type = case_when(
    predict < 0.2 ~ "cat",
    predict > 0.9 ~ "dog",
    is.na(predict) ~ "no photo",
    TRUE ~ "other"
  ),
  animal_text = case_when(
    grepl("cat|kitten|Cat|Kitten", text) ~ "cat",
    grepl("dog|pup|Dog|Pup", text) ~ "dog",
    TRUE ~ "other"
  )) %>%
  filter(is_quote == F, is_retweet == F)


tweets_pred %>%
  filter(animal_type %in% c("cat","dog")) %>%
  # filter(retweet_count < 200) %>%
  gather(metric, count, favorite_count:retweet_count) %>%
  mutate(metric = gsub("_count","",metric)) %>%
  ggplot(aes(x=metric, y=count, fill = animal_type)) +
  geom_point(pch=21, size=1,
```

```
            position = position_jitterdodge(jitter.width = 0.2, jitter.height = 0)) +
  geom_boxplot(alpha = 0.5, outlier.shape = NA) +
  scale_y_continuous(trans = "log2") +
  # ylim(c(0,100)) +
  ylab("Count") + xlab("") +
  ggtitle("Tweets with cat photos \nare retweeted more")
```




```
tweets_pred %>%
  # filter(animal_type %in% c("cat","dog")) %>%
  # filter(retweet_count < 200) %>%
  gather(metric, count, favorite_count:retweet_count) %>%
  mutate(metric = gsub("_count","",metric)) %>%
  ggplot(aes(x=metric, y=count, fill = animal_text)) +
  geom_point(pch=21, size=1,
            position = position_jitterdodge(jitter.width = 0.2, jitter.height = 0)) +
  geom_boxplot(alpha = 0.5, outlier.shape = NA) +
  # ylim(c(0,100)) +
  scale_y_continuous(trans = "log2") +
  ylab("Count") + xlab("") +
  ggtitle("Tweets with cat text \nare retweeted more")
```



Conclusions and Next Steps

Keep on tweeting! If someone wanted to dig deeper into this data, they could:

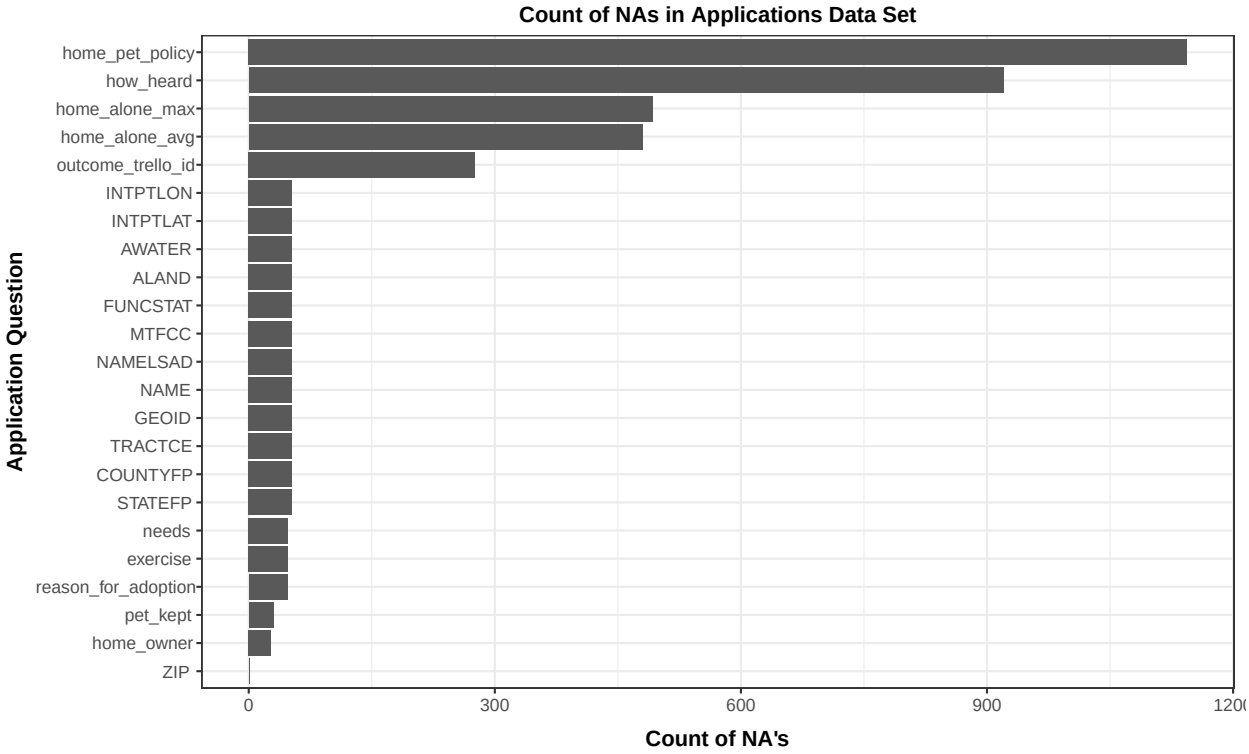
- Perform better image analysis on photos tweeted
- Try to link tweets about certain topics (a specific animal, request for donations) to specific outcomes (did the animal get adopted? did they receive more donations?)

5. Data considerations

Data Issues affecting Analyses

Missing Data

Overall we were able to achieve some insights given the application data. However, we were at times limited due to missing data in the applications data set. Below is a plot that shows counts of `NA`'s in each column of the data set.



The question with the most missing data is one regarding the home pet policy. This seems like an important question, especially for renters, and a non-response here may require manual follow up by PAWS staff. Making this a required question could save some time in the future.

Unlimited Responses and Response Validation

Like many of the other teams, we ran into several challenges as a result of questions having a wide range of possible responses and illogical answers. For example, the 12 different responses below are for the Allergy question:

Response	Count
no-allergies	1,694
mildly-allergic	130
not-sure	38
not-sure,no-allergies	16
very-allergic	10
no-allergies,mildly-allergic	5
no-allergies,not-sure	5
mildly-allergic,no-allergies	3

Response	Count
mildly-allergic,very-allergic	3
mildly-allergic,not-sure	1
very-allergic,mildly-allergic	1
very-allergic,no-allergies	1

In one case the responses conflict with each other: "very-allergic,no-allergies". This make grouping the data after the fact almost impossible because its not clear if this applicant has allergies or not. This is one example, but there were some other cases where this problem occurred as well, such as for the questions relating to Experience and Where the Pet Will be Kept.

For the monthly budget question, there were several negative numbers and some extremely large, strange values (i.e $150,159.00). Utilizing some kind of response validation logic (i.e.only allow positive values) and limiting the range of responses to a reasonable size given the question (in this case maybe between 200 and 1,000) would also make future analysis much more efficient.

Recommendations for Collecting Clean Data

One of the most important recommendations moving forward would be to redesign the application to enforce standardized, limited and logical responses. Allowing only a single response combined with a limited response set would make analysis much easier in the future. Doing so will save PAWS staff time when reviewing applications *and* make future analyses easier and can lead to better insights.

Conclusions and Next Steps

This section should have a bulletpoint list of what conclusions can be drawn from the analyses that were performed, and what next steps should be taken, both by PAWS and by R-Ladies.