Towards A Clear Understanding Of Rural Internet: What Statistical Measures Can Be Used To Assess, Compare And Forecast Internet Speeds For Rural Canadian Communities?

A Consulting Project for Math 6627 (1/3)

Benjamin Smith

10 February 2022

Introduction

(Quoted from the SCC website)

The Government of Canada has committed to helping 95% of Canadian households and businesses access high-speed internet at minimum speeds of 50 Mbps download and 10 Mbps upload (hereinafter referred to as the "Commitment") by 2026, and 100% by 2030. According to the CRTC, currently 45.6% of rural community households have access to the Commitment based on what's available to them via an Internet Service Provider (e.g. Shaw, Telus, etc.) in their region, rather than what a rural household actually realizes at home in terms of internet speeds.

For this case study, the SCC would like to understand the state of internet connectivity in both rural and underserved Canadian communities using consumer-provided data. The SCC claims that by using data directly from the consumer, it is possible to better understand connectivity in these communities as measured by the consumers in their own homes.

Specifically, the following is desired:

1. A statistical analysis of the current realized and forecasted internet speeds (upload and download) for rural and undeserved communities in terms of progress towards the Commitment;
2. A comparative analysis of rural and underserved communities in terms of progress towards the Commitment; and
3. The identification of statistically reliable methods to assess and compare rural and underserved communities's realized internet access.For this study in particular, the identification of reliable and reproducible statistical methods to understand connectivity of rural and underserved Canadian communities is critical.

The following analysis aims to address the above in a practical and concise manner.

The Data

The data was made available by the Statiscal Society of Canada with Ookla and Statistics Canada. One of the first things to check regarding the data is to see if any missing data is present in the dataset. Figure 1 shows that most of the missing data is related to population center information. Namely data on population center id, type and class (PCUID, PCTYPE, PCLASS).

```
library(tidyverse)
library(ggthemes)
```

```r
library(ggspatial)
library(plotly)
library(rnaturalearth)
library(sf)
library(scales)
library(reshape2)
#dt<- readr::read_csv("./ConsultingData/ookla-canada-speed-tiles.csv")
# Accomidating for mtor
setwd("/home/ben2908")
dt<- readr::read_csv("./ookla-canada-speed-tiles.csv")
```

```r
naniar::vis_miss(dt ,warn_large_data = FALSE)+
  theme(axis.text.x=element_text(angle=90))
```

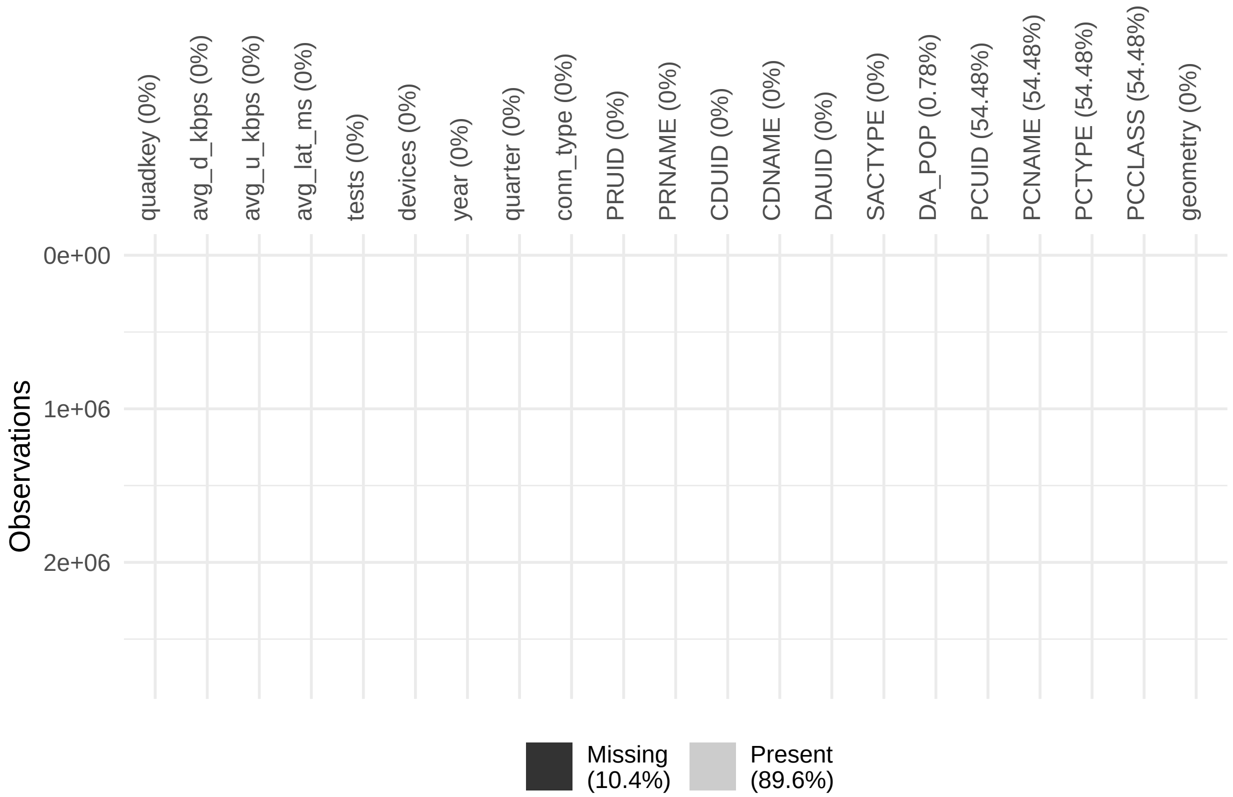


Figure 1: Missing data present in dataset provied

While it is possible to apply some treatment to the missing data, after removal, the remaining sample is 1,252,560 rows, which is still usable for this analysis. As such a filtered data set is used.

Analysis

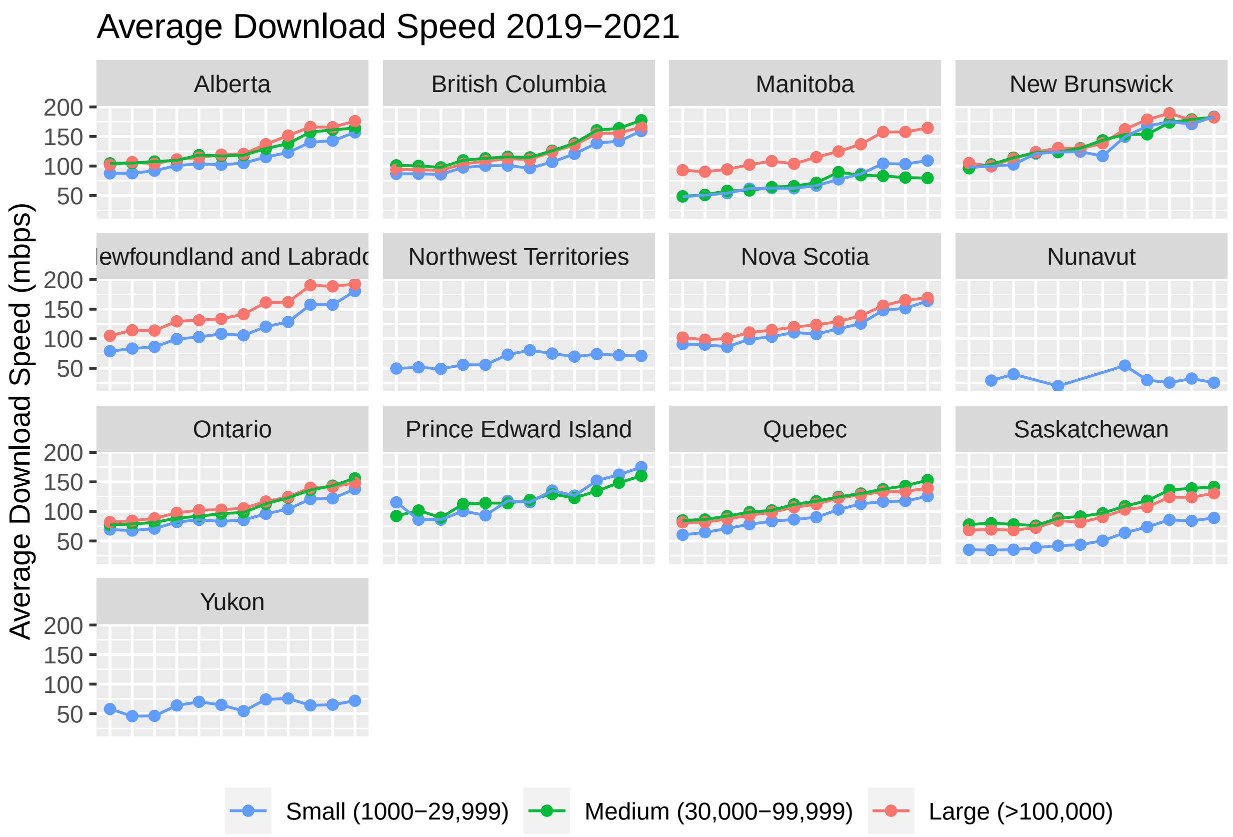
Rural and Urban Towns

Figures 2 and 3 show that on average, most provinces are keeping to the Commitment above and beyond the requirements provided for all communities. The provinces which appear to be experiencing challenges with this are the Northwest Territories, Nunavut and Yukon- all of whom have available data on small population centers[1]. While this does offer a "birds eye view" a more accurate portrayal is to look at the proportion of tiles in population centers which are meeting the agreement and which are not. Figures 4-9 outline such characteristics. In terms of small population centers, all provinces appear to be making progress in the Comittment with the exception to Nunavut

```
# Need to filter based on PCCLASS
dt %>%
  filter(!is.na(PCCLASS)) %>%
  group_by(PCCLASS,PRNAME,year,quarter) %>%
  summarize(average_d_time = mean(avg_d_kbps/1000)) %>%
  mutate(year_quarter=paste(year,quarter),
         Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
         popcenter_class = ifelse(PCCLASS==2, "Small (1000-29,999)",ifelse(PCCLASS==3, "Medium (30,000-9
  ggplot(mapping=aes(x= year_quarter,y=average_d_time,color= popcenter_class,group= popcenter_class))+
  geom_point()+
  geom_line()+
  facet_wrap(~Province)+
  ggtitle("Average Download Speed 2019-2021")+
  labs(y="Average Download Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title.x = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```

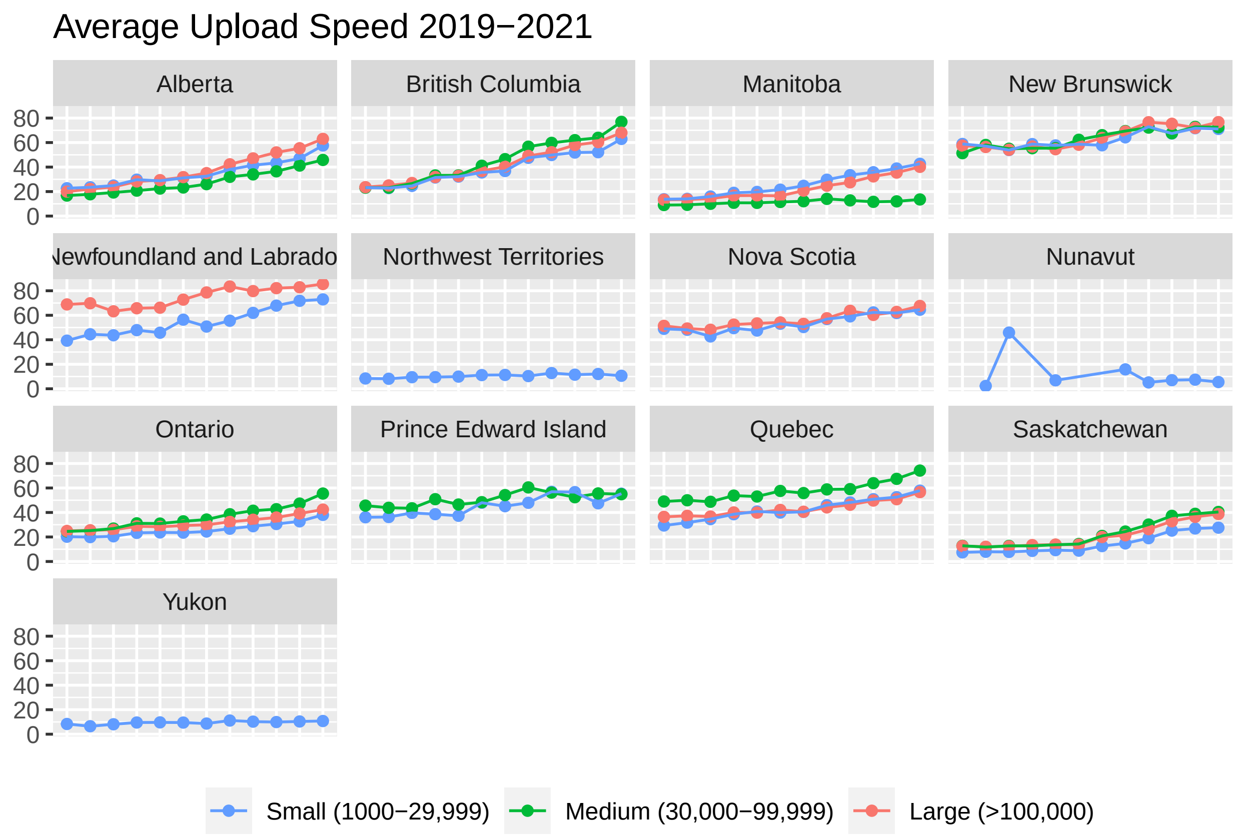
```
## `summarise()` has grouped output by 'PCCLASS', 'PRNAME', 'year'. You can override using the `.groups
```

[1] According to Wikipedia there are only small population centers in the Canadian Territories as of 2016. Reference: https://en.wikipedia.org/wiki/List_of_population_centres_in_the_Canadian_Territories

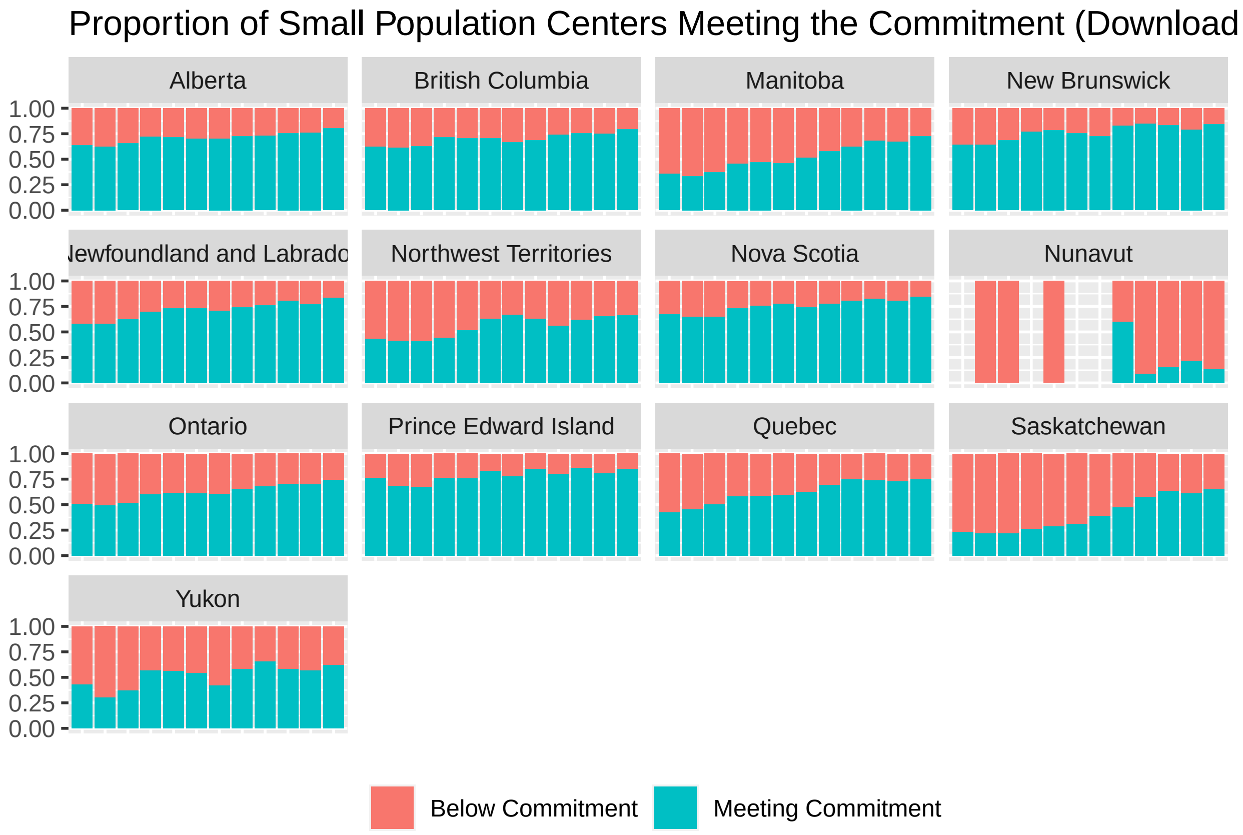


```
# Need to filter based on PCCLASS
dt %>%
  filter(!is.na(PCCLASS)) %>%
  group_by(PCCLASS,PRNAME,year,quarter) %>%
  summarize(average_u_time = mean(avg_u_kbps/1000)) %>%
  mutate(year_quarter=paste(year,quarter),
         Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
         popcenter_class = ifelse(PCCLASS==2, "Small (1000-29,999)",ifelse(PCCLASS==3, "Medium (30,000-
  ggplot(mapping=aes(x= year_quarter,y=average_u_time,color= popcenter_class,group= popcenter_class))+
  geom_point()+
  geom_line()+
  facet_wrap(~Province)+
  ggtitle("Average Upload Speed 2019-2021")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```

```
## `summarise()` has grouped output by 'PCCLASS', 'PRNAME', 'year'. You can override using the `.groups
```

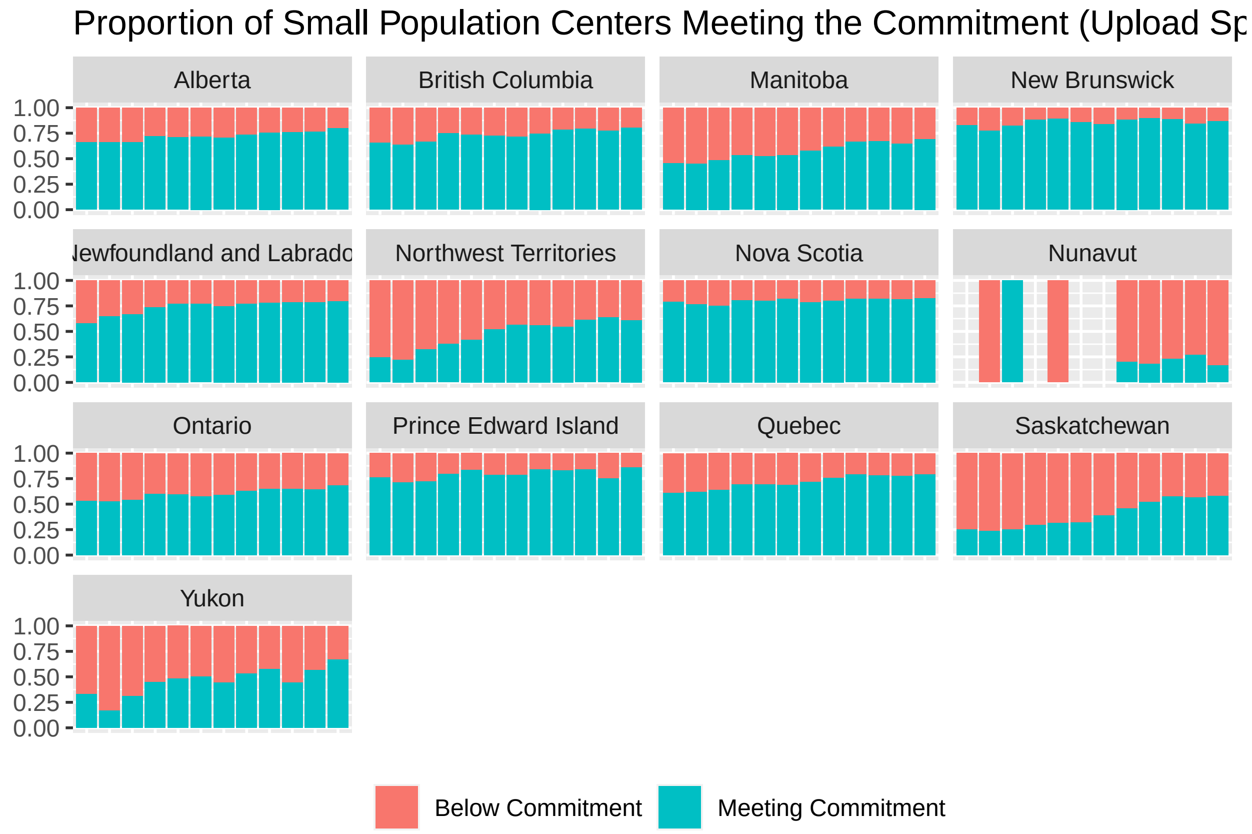


```
# Small Provinces

smallProvs_Download<- dt %>%
              mutate(year_quarter=paste(year,quarter),
                     Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
                      popcenter_class = ifelse(PCCLASS==2,
                                               "Small (1000-29,999)",
                                               ifelse(PCCLASS==3,
                                                      "Medium (30,000-99,999)",
                                                      ifelse(PCCLASS==4,"Large (>100,000)",
                                                             "OTHER"))),
                      d_status=ifelse((avg_d_kbps/1000)<50,"Below Commitment","Meeting Commitment"),
                      u_status=ifelse((avg_u_kbps/1000)<10,"Below Commitment","Meeting Commitment")) %>
                filter(!is.na(PCCLASS),popcenter_class=="Small (1000-29,999)") %>%
                group_by(Province, year_quarter) %>%
                count(d_status)


smallProvs_Download<- do.call(rbind,
                     by(smallProvs_Download,smallProvs_Download["Province"],tibble) %>%
                     lapply(function(x) do.call(rbind,by(x,x["year_quarter"],tibble) %>%
                                                  lapply(function(y) y %>% mutate(prop = y$n/sum(y$n)))
)

smallProvs_Download %>%
  ggplot(mapping=aes(x=year_quarter,y=prop,fill=d_status))+
  geom_bar(stat="identity")+
  facet_wrap(~Province)+
  ggtitle("Proportion of Small Population Centers Meeting the Commitment (Download Speed)")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```

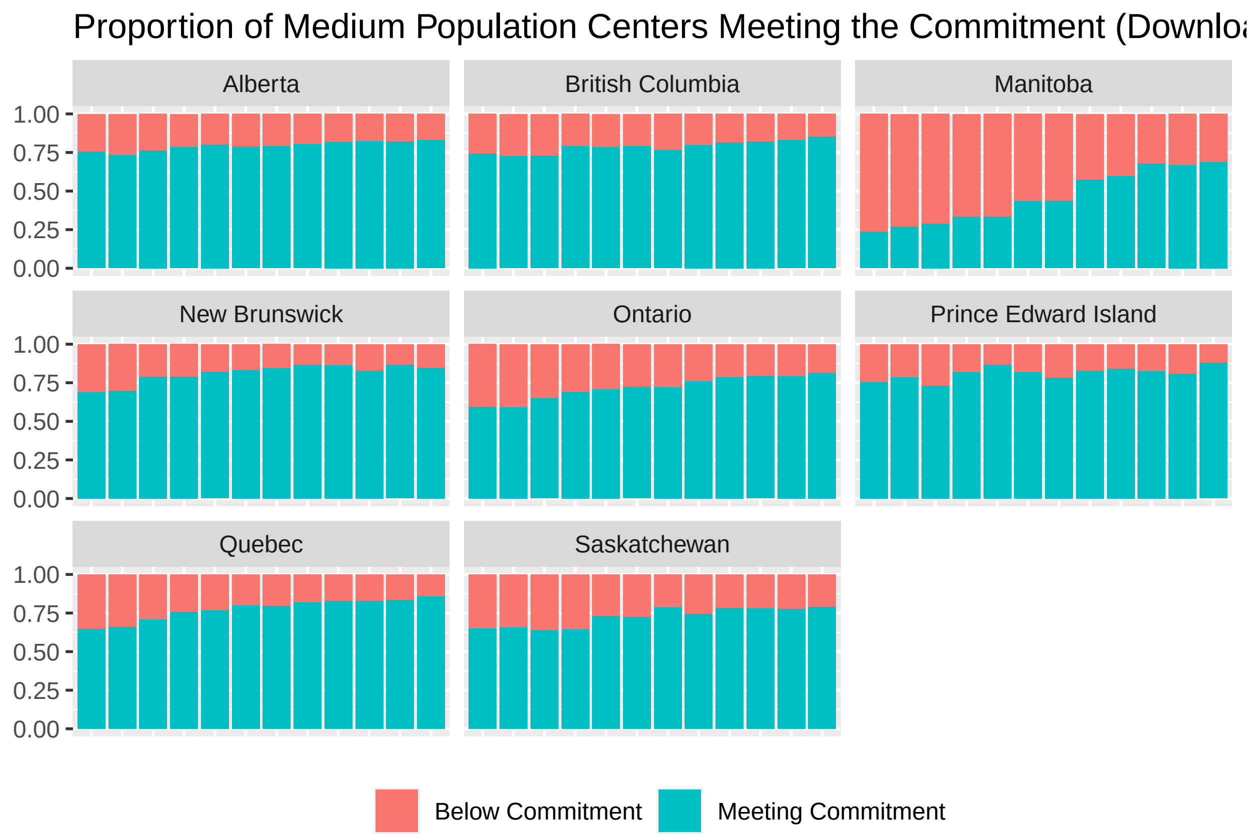


```
smallProvs_Upload<- dt %>%
              mutate(year_quarter=paste(year,quarter),
                     Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
                      popcenter_class = ifelse(PCCLASS==2,
                                               "Small (1000-29,999)",
                                               ifelse(PCCLASS==3,
                                                      "Medium (30,000-99,999)",
                                                      ifelse(PCCLASS==4,"Large (>100,000)",
                                                             "OTHER"))),
                      d_status=ifelse((avg_d_kbps/1000)<50,"Below Commitment","Meeting Commitment"),
                      u_status=ifelse((avg_u_kbps/1000)<10,"Below Commitment","Meeting Commitment")) %>
                filter(!is.na(PCCLASS),popcenter_class=="Small (1000-29,999)") %>%
                group_by(Province, year_quarter) %>%
                count(u_status)
```

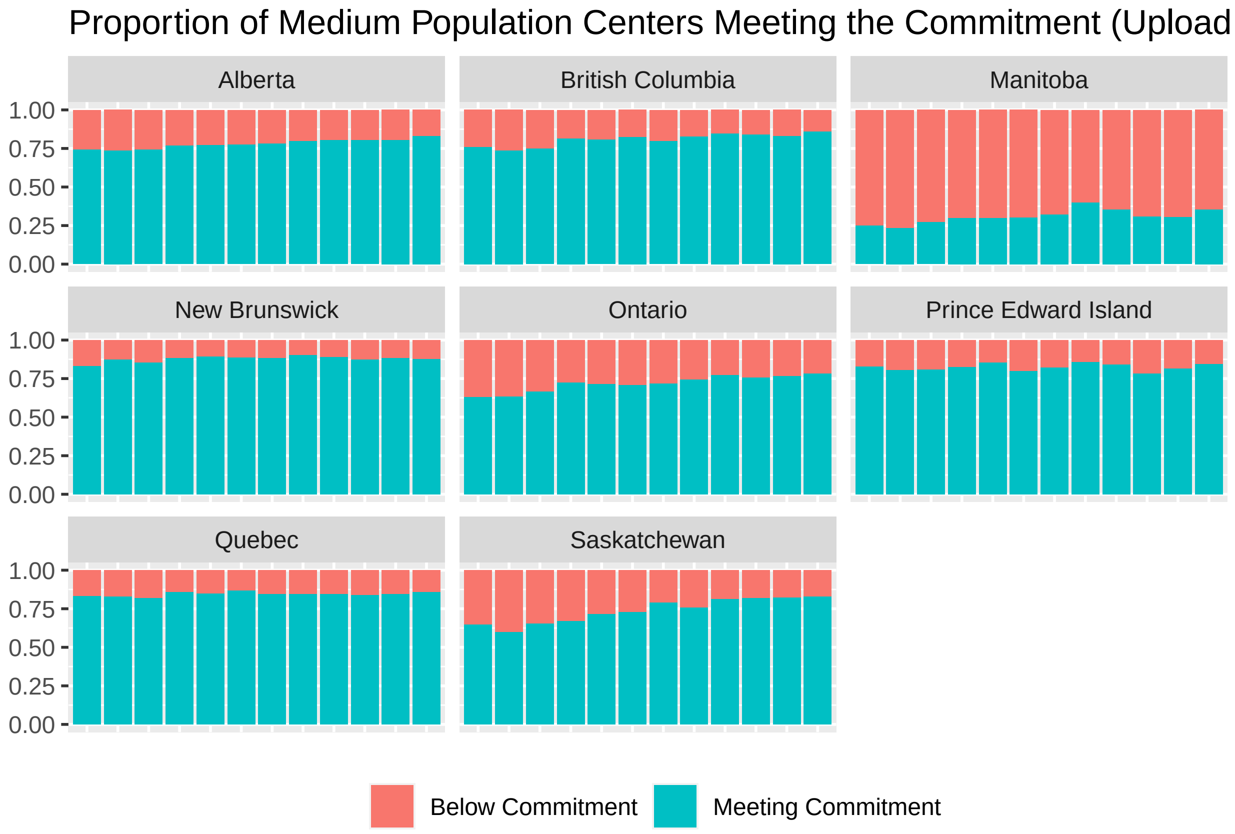
```
smallProvs_Upload<- do.call(rbind,
                     by(smallProvs_Upload,smallProvs_Upload["Province"],tibble) %>%
                     lapply(function(x) do.call(rbind,by(x,x["year_quarter"],tibble) %>%
                                                  lapply(function(y) y %>% mutate(prop = y$n/sum(y$n)))
)

smallProvs_Upload %>%
  ggplot(mapping=aes(x=year_quarter,y=prop,fill=u_status))+
  geom_bar(stat="identity")+
  facet_wrap(~Province)+
  ggtitle("Proportion of Small Population Centers Meeting the Commitment (Upload Speed)")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```



```
# Medium
medProvs_Download<- dt %>%
              mutate(year_quarter=paste(year,quarter),
                     Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
                      popcenter_class = ifelse(PCCLASS==2,
                                               "Small (1000-29,999)",
                                               ifelse(PCCLASS==3,
                                                      "Medium (30,000-99,999)",
                                                      ifelse(PCCLASS==4,"Large (>100,000)",
                                                             "OTHER"))),
                      d_status=ifelse((avg_d_kbps/1000)<50,"Below Commitment","Meeting Commitment"),
                      u_status=ifelse((avg_u_kbps/1000)<10,"Below Commitment","Meeting Commitment")) %>%
                filter(!is.na(PCCLASS),popcenter_class=="Medium (30,000-99,999)") %>%
                group_by(Province, year_quarter) %>%
                count(d_status)


medProvs_Download<- do.call(rbind,
                     by(medProvs_Download,medProvs_Download["Province"],tibble) %>%
                     lapply(function(x) do.call(rbind,by(x,x["year_quarter"],tibble) %>%
                                                  lapply(function(y) y %>% mutate(prop = y$n/sum(y$n)))
)

medProvs_Download %>%
  ggplot(mapping=aes(x=year_quarter,y=prop,fill=d_status))+
  geom_bar(stat="identity")+
  facet_wrap(~Province)+
  ggtitle("Proportion of Medium Population Centers Meeting the Commitment (Download Speed)")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```

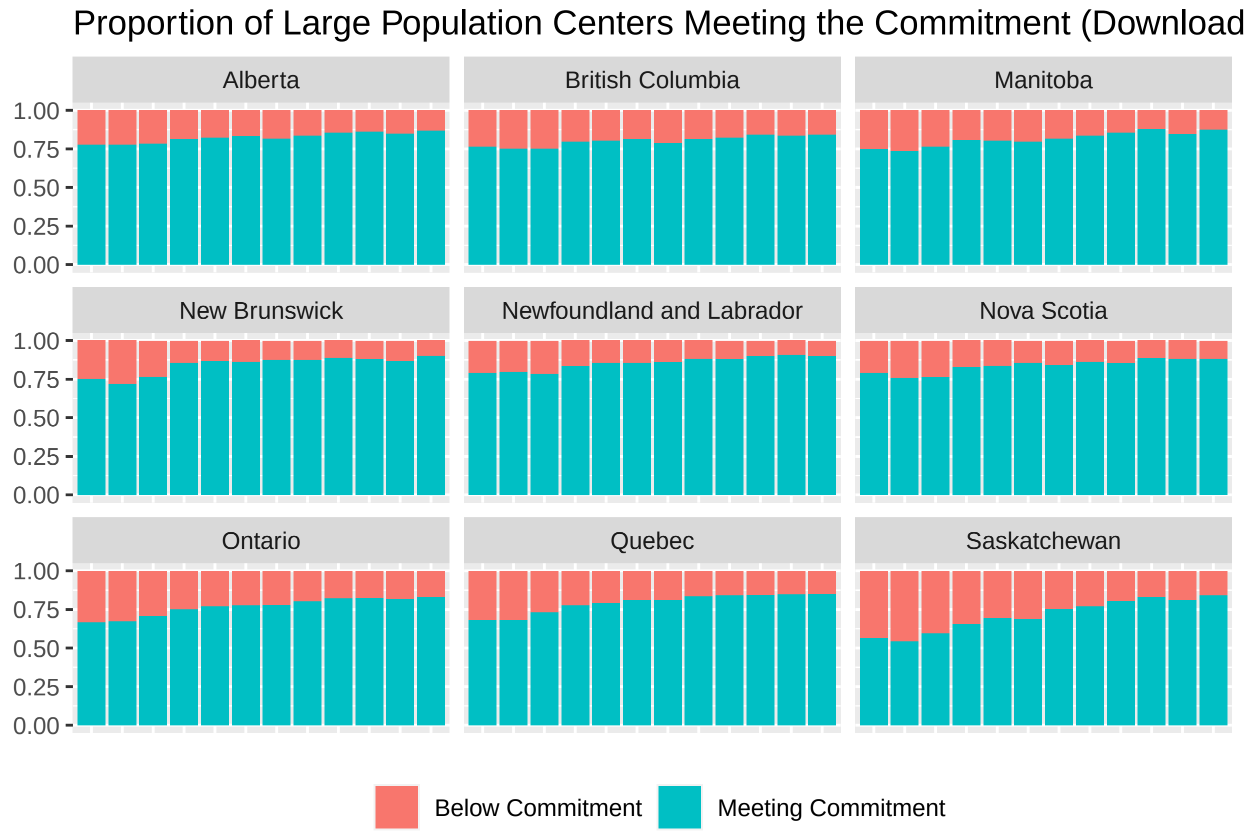


```
# Medium
medProvs_Upload<- dt %>%
              mutate(year_quarter=paste(year,quarter),
                     Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
                      popcenter_class = ifelse(PCCLASS==2,
                                               "Small (1000-29,999)",
                                               ifelse(PCCLASS==3,
                                                      "Medium (30,000-99,999)",
                                                      ifelse(PCCLASS==4,"Large (>100,000)",
                                                             "OTHER"))),
                      d_status=ifelse((avg_d_kbps/1000)<50,"Below Commitment","Meeting Commitment"),
                      u_status=ifelse((avg_u_kbps/1000)<10,"Below Commitment","Meeting Commitment")) %>
                filter(!is.na(PCCLASS),popcenter_class=="Medium (30,000-99,999)") %>%
                group_by(Province, year_quarter) %>%
                count(u_status)


medProvs_Upload<- do.call(rbind,
                     by(medProvs_Upload,medProvs_Upload["Province"],tibble) %>%
                     lapply(function(x) do.call(rbind,by(x,x["year_quarter"],tibble) %>%
                                                  lapply(function(y) y %>% mutate(prop = y$n/sum(y$n)))
)

medProvs_Upload %>%
  ggplot(mapping=aes(x=year_quarter,y=prop,fill=u_status))+
  geom_bar(stat="identity")+
  facet_wrap(~Province)+
  ggtitle("Proportion of Medium Population Centers Meeting the Commitment (Upload Speed)")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```

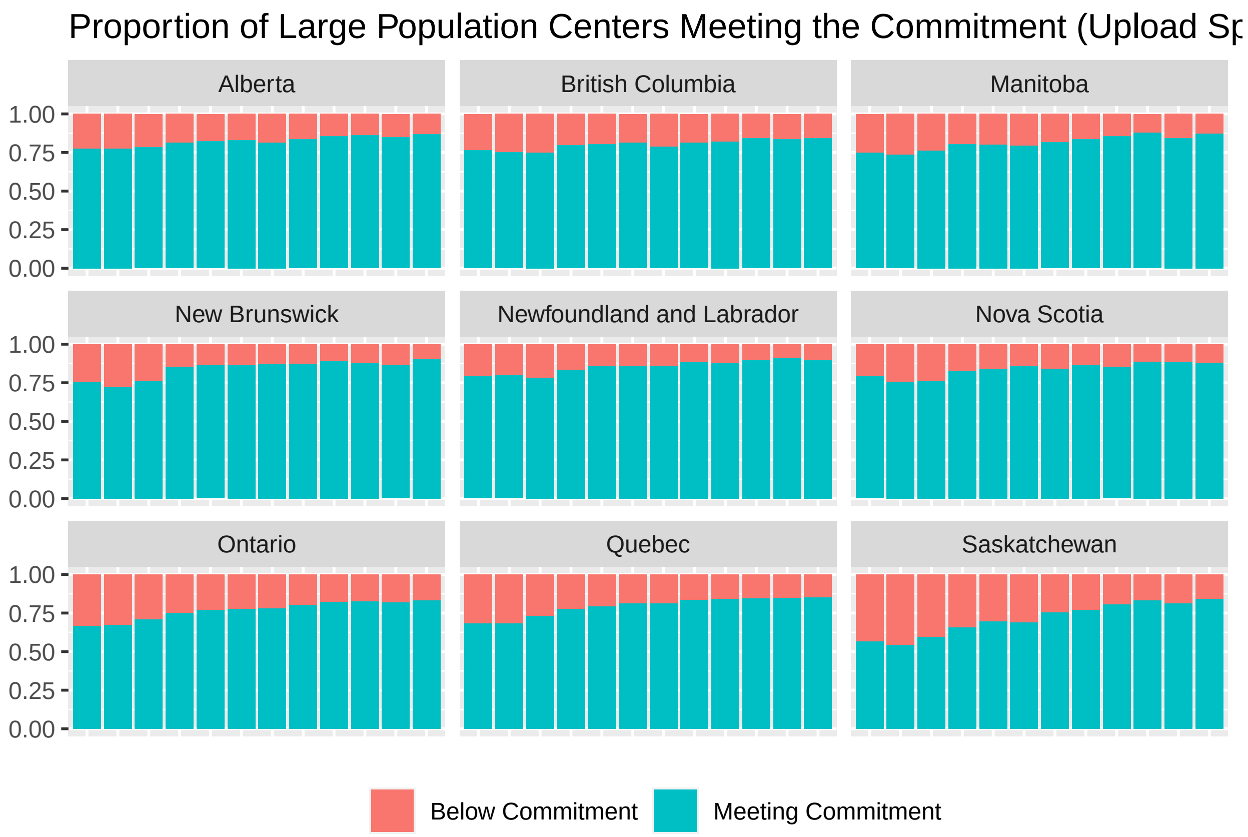


```
# Large
largeProvs_Download<- dt %>%
              mutate(year_quarter=paste(year,quarter),
                     Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
                      popcenter_class = ifelse(PCCLASS==2,
                                               "Small (1000-29,999)",
                                               ifelse(PCCLASS==3,
                                                      "Medium (30,000-99,999)",
                                                      ifelse(PCCLASS==4,"Large (>100,000)",
                                                             "OTHER"))),
                      d_status=ifelse((avg_d_kbps/1000)<50,"Below Commitment","Meeting Commitment"),
                      u_status=ifelse((avg_u_kbps/1000)<10,"Below Commitment","Meeting Commitment")) %>
                filter(!is.na(PCCLASS),popcenter_class=="Large (>100,000)") %>%
                group_by(Province, year_quarter) %>%
                count(d_status)
```

```
largeProvs_Download<- do.call(rbind,
                     by(largeProvs_Download,largeProvs_Download["Province"],tibble) %>%
                     lapply(function(x) do.call(rbind,by(x,x["year_quarter"],tibble) %>%
                                                  lapply(function(y) y %>% mutate(prop = y$n/sum(y$n)))
)

largeProvs_Download %>%
  ggplot(mapping=aes(x=year_quarter,y=prop,fill=d_status))+
  geom_bar(stat="identity")+
  facet_wrap(~Province)+
  ggtitle("Proportion of Large Population Centers Meeting the Commitment (Download Speed)")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```

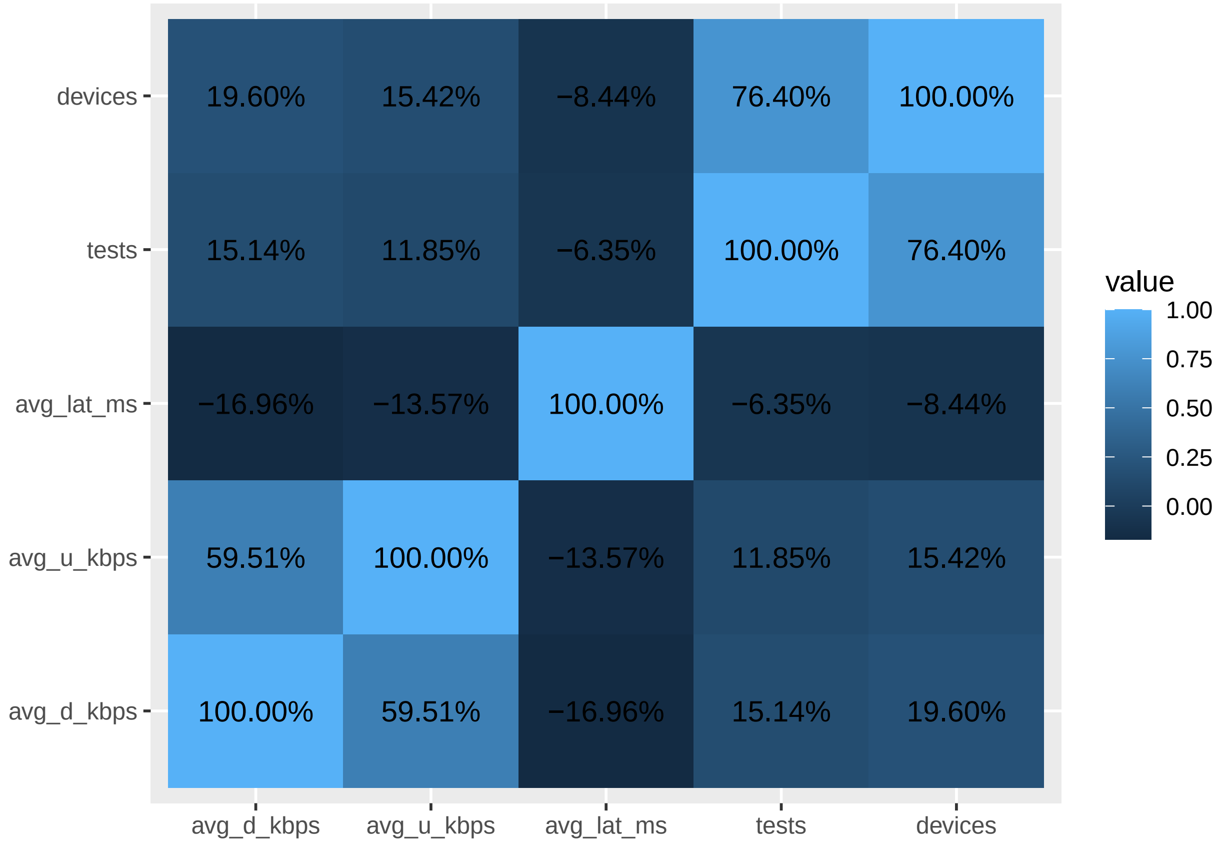


```
# Large
largeProvs_Upload<- dt %>%
              mutate(year_quarter=paste(year,quarter),
                     Province= ifelse(grepl("\\/",PRNAME),PRNAME %>% str_extract('.*(?= \\/)'),PRNAME),
                      popcenter_class = ifelse(PCCLASS==2,
                                               "Small (1000-29,999)",
                                               ifelse(PCCLASS==3,
                                                      "Medium (30,000-99,999)",
                                                      ifelse(PCCLASS==4,"Large (>100,000)",
                                                             "OTHER"))),
                      d_status=ifelse((avg_d_kbps/1000)<50,"Below Commitment","Meeting Commitment"),
                      u_status=ifelse((avg_u_kbps/1000)<10,"Below Commitment","Meeting Commitment")) %>
                filter(!is.na(PCCLASS),popcenter_class=="Large (>100,000)") %>%
                group_by(Province, year_quarter) %>%
                count(d_status)


largeProvs_Upload<- do.call(rbind,
                     by(largeProvs_Upload,largeProvs_Upload["Province"],tibble) %>%
                     lapply(function(x) do.call(rbind,by(x,x["year_quarter"],tibble) %>%
                                                  lapply(function(y) y %>% mutate(prop = y$n/sum(y$n)))
)

largeProvs_Upload %>%
  ggplot(mapping=aes(x=year_quarter,y=prop,fill=d_status))+
  geom_bar(stat="identity")+
  facet_wrap(~Province)+
  ggtitle("Proportion of Large Population Centers Meeting the Commitment (Upload Speed)")+
  labs(y="Average Upload Speed (mbps)")+
  scale_color_discrete(guide = guide_legend(reverse=TRUE))+
  theme(axis.ticks.x = element_blank(),
        axis.text.x = element_blank(),
        axis.title = element_blank(),
        legend.title=element_blank(),
        legend.position = "bottom")
```



Correlation Between Upload, Download and Latency times

```
dt %>%
  dplyr::select(avg_d_kbps,avg_u_kbps, avg_lat_ms,tests,devices) %>%
  cor() %>%
  melt() %>%
  ggplot(mapping=aes(x=Var2,y=Var1,fill=value))+
  geom_tile()+
  scale_fill_gradient()+
  geom_text(mapping=aes(label=scales::percent(value)))+
  theme(axis.title = element_blank())
```



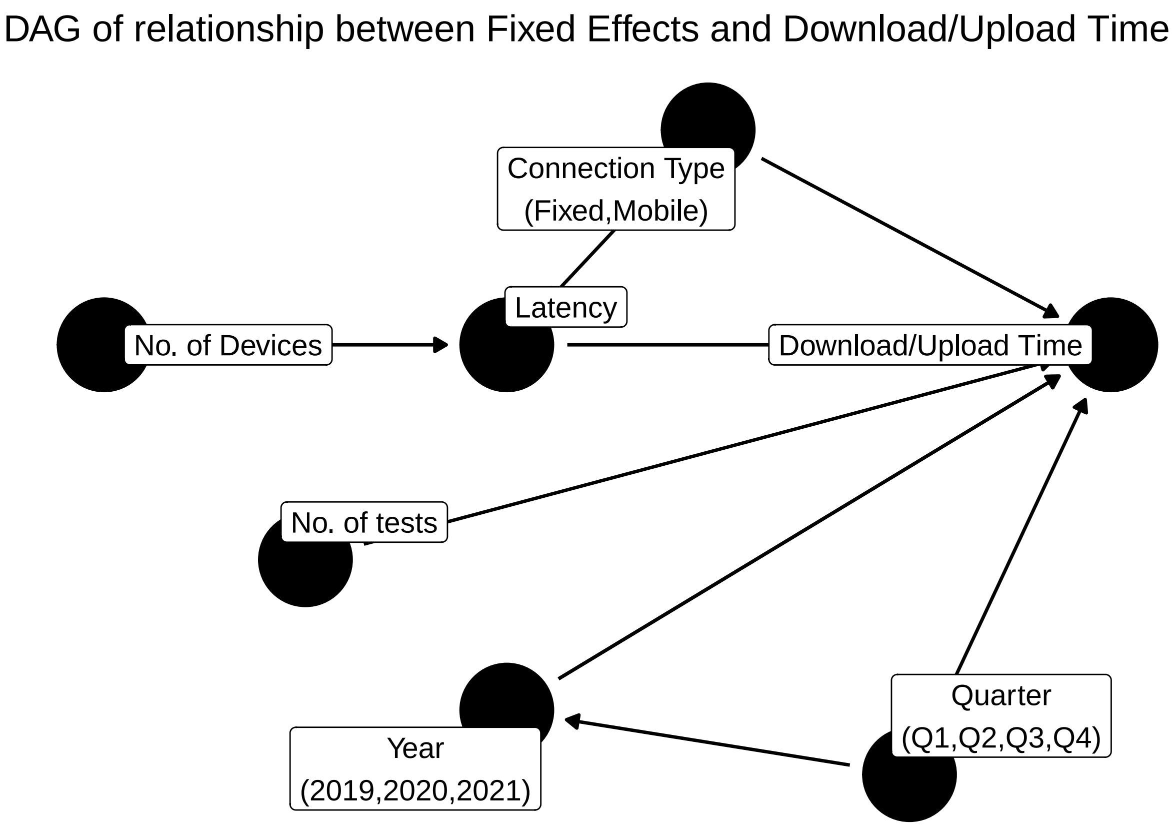
Current Realized And Forecasted Internet Speeds (Upload And Download)

Due to the large number of observations, approaching this dataset with standard visuals was not easy to do. In lieu of this the use of directed acyclic graphs (DAGs) is employed to understand the relationship between variables available and develop the model accordingly.

```
library(ggdag)
```

```
##
## Attaching package: 'ggdag'

## The following object is masked from 'package:stats':
##
##     filter
```

```
dagify(download_time~latency,
       download_time ~ conn_type,
       download_time ~tests,
       download_time~year,
       download_time~quarter,
       quarter~year,
       latency~ conn_type,
       latency~no_of_devices,
       year~quarter,
       labels=c("download_time"="Download/Upload Time",
                "latency" = "Latency",
                "conn_type"="Connection Type\n(Fixed,Mobile)",
                "no_of_devices"="No. of Devices",
                "tests"="No. of tests",
                "year"="Year\n(2019,2020,2021)",
                "quarter"="Quarter\n(Q1,Q2,Q3,Q4)")) %>%
  tidy_dagitty() %>%
  mutate(xend=c(10.5,9,10.5,9,10.5,9,10.5,10.5,10.5,NA),
         yend=c(0,0,0,0,0,0,-1.7,0,0,0,NA),
         x=ifelse(name=="download_time",10.5,
                  ifelse(name=="latency",9,
                         ifelse(name=="conn_type",9.5,
                                       ifelse(name=="tests",8.5,
                                              ifelse(name=="year",9,
                                                     ifelse(name=="quarter",10,
                                                            8)))))),
         y=ifelse(name=="download_time",0,
                  ifelse(name=="latency",0,
                         ifelse(name=="conn_type",1,
                                       ifelse(name=="tests",-1,
                                              ifelse(name=="year",-1.7,
                                                     ifelse(name=="quarter",-2,
                                                            0)))))),
         effectType=ifelse(name %in% c("download_time",
                "latency",
                "conn_type",
                "no_of_devices",
                "tests",
                "year",
                "quarter"),"Fixed","Random")) %>%
  ggdag(text=FALSE,use_labels = "label")+
  ggtitle("DAG of relationship between Fixed Effects and Download/Upload Time")+
  theme_dag()
```



It is from this DAG that the following mixed model is constructed.

$$Y = X\beta + Zb$$

Where X is the design matrix for the fixed effects, Z is the design matrix of the random effects and β and b are the fixed and random effects vectors. The fixed and random effects are:

Fixed Effects	Random Effect
No. of devices	Province
Connection Type	Population Center Class
No. of Tests	Province*Population Center Class
Year	
Quarter	
Year*Quarter	

After running this model in SAS note the following:

```r
X_download<- model.matrix(avg_d_kbps~ devices + conn_type+ tests + year*quarter + PRNAME,data=dt %>%
  dplyr::select(avg_d_kbps,PRNAME, devices, conn_type, tests, year,quarter))
```

```r
library(lme4)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
fit_download_time <- lmer(avg_d_kbps~ devices + conn_type+ tests + year*quarter+(1|PRNAME)+(1|PCCLASS)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```
summary(fit_download_time)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: avg_d_kbps ~ devices + conn_type + tests + year * quarter + (1 |
##     PRNAME) + (1 | PCCLASS) + (1 | PRNAME:PCCLASS)
##    Data: data.frame(dt %>% filter(!is.na(PCCLASS)), stringsAsFactors = T)
##
## REML criterion at convergence: 31763418
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.7386 -0.6671 -0.1522  0.4635 11.8618
##
## Random effects:
##  Groups         Name        Variance  Std.Dev.
##  PRNAME:PCCLASS (Intercept) 1.289e+08 11353
##  PRNAME         (Intercept) 8.370e+08 28931
##  PCCLASS        (Intercept) 1.358e+08 11654
##  Residual                   6.035e+09 77686
## Number of obs: 1252560, groups:  PRNAME:PCCLASS, 30; PRNAME, 13; PCCLASS, 3
##
## Fixed effects:
##                   Estimate Std. Error t value
## (Intercept)     -4.383e+07  3.511e+05 -124.83
## devices          3.670e+02  1.057e+01   34.70
## conn_typemobile -2.840e+04  1.661e+02 -171.02
## tests            3.849e+01  2.293e+00   16.78
## year             2.174e+04  1.737e+02  125.17
## quarterQ2       -1.362e+07  4.963e+05  -27.45
## quarterQ3       -1.205e+07  4.875e+05  -24.73
## quarterQ4       -1.346e+07  4.889e+05  -27.53
## year:quarterQ2   6.747e+03  2.457e+02   27.46
## year:quarterQ3   5.972e+03  2.413e+02   24.75
## year:quarterQ4   6.672e+03  2.420e+02   27.57
##
## Correlation of Fixed Effects:
##             (Intr) devics cnn_ty tests  year   qrtrQ2 qrtrQ3 qrtrQ4 yr:qQ2
## devices      0.004
## conn_typmbl  0.029  0.190
## tests        0.026 -0.733  0.060
## year        -1.000 -0.004 -0.029 -0.026
## quarterQ2   -0.706  0.008 -0.004 -0.016  0.706
## quarterQ3   -0.720 -0.002 -0.020 -0.020  0.720  0.508
## quarterQ4   -0.718 -0.019 -0.024 -0.019  0.719  0.507  0.517
## year:qrtrQ2  0.706 -0.008  0.004  0.016 -0.706 -1.000 -0.508 -0.507
## year:qrtrQ3  0.720  0.002  0.020  0.020 -0.720 -0.508 -1.000 -0.517  0.508
## year:qrtrQ4  0.718  0.019  0.024  0.019 -0.719 -0.507 -0.517 -1.000  0.507
##             yr:qQ3
## devices
## conn_typmbl
## tests
## year
## quarterQ2
## quarterQ3
## quarterQ4
## year:qrtrQ2
## year:qrtrQ3
## year:qrtrQ4  0.517
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

Rural and Underserved communities in terms of progress towards the Commitment

The identification of statistically reliable methods to assess and compare rural and underserved communities's realized internet access.

- Test test test.

Maps

```r
# Convert dataset to sf object
dtt<- dt %>% st_as_sf(wkt  = "geometry")
```

```r
world_map <- ne_countries(scale = "large", returnclass = 'sf')
canada_map <- world_map %>% filter(name =="Canada")
```

```r
# dtt %>%
#   ggplot()+
#   geom_sf()
```

```r
# Loading images
setwd("/home/ben2908")

df<- read_sf('ookla-canada-speed-tiles.shp')

df2<- read_sf('lpr_000b16a_e.shp')
```

```r
# ggplot()+
#   geom_sf(data=df2)+
#   geom_sf(data=df,mapping=aes(color=avg_d_kbps))+
#   ggtitle("Average Download Times in Canada")+
#   facet_wrap(year~quarter)
```

```r
# ggplot()+
#   geom_sf(data=df2)+
#   geom_sf(data=df,mapping=aes(color=avg_u_kbps))+
#   ggtitle("Average Upload Times in Canada")+
#   facet_wrap(year~quarter)
# ggplot()+
#   geom_sf(data=df2)+
#   geom_sf(data=df,mapping=aes(color=avg_lat_ms))+
#   ggtitle("Average Latency Times in Canada")+
#   facet_wrap(year~quarter)
```

References

Code Appendix

SAS Code

```
/*Update File Path Accordingly*/
FILENAME REFFILE '.../ookla-canada-speed-tiles.csv';
PROC IMPORT DATAFILE=REFFILE
    DBMS=CSV
    OUT=DT;
    GETNAMES=YES;
RUN;

PROC CONTENTS DATA=DT;
RUN;

/*Download Time*/
proc mixed data=DT method=reml covtest;
class PRNAME quarter year conn_type;
model avg_d_kbps= devices conn_type tests year quarter quarter*year;
random PRNAME/s;
run;


/*Upload Time*/
proc mixed data=DT method=reml covtest;
class PRNAME quarter year conn_type;
model avg_u_kbps= devices conn_type tests year quarter quarter*year;
random PRNAME/s;
run;
```