October 7, 2022

The results below are generated from an R script.

```r
#' # Load necessary packages
# Load Necessary packages ------------------------------------------------------
library(pacman)

p_load(gapminder,dplyr, tidyr, stringr, readxl, writexl, ggplot2, forcats,
       sjlabelled,
       here, hablar, summarytools, splitstackshape, ggrepel,quantreg,
       scales,tidylog)

library(dplyr)
library(quantreg)
library(scales)
library(tidyr)
library(tidylog, warn.conflicts = FALSE)

# IMPORT DATA  --------------------------------------------------------
pv_database <- read_xlsx(here::here("data", "pv_database_costing_v3.xlsx"),
                         col_types = "text")

minigrids_database <- read_xlsx(here::here("data", "system_sizing.xlsx"),
                                col_types = "text")

pv_database <- pv_database %>% mutate(total_cost_irs=as.numeric(total_cost_irs))
pv_database <- pv_database %>% mutate_at(c(9:13,16:26),as.numeric)

# SPIS - Solar Pump Irrigation Systems -----------------------------------------

spis_database <- pv_database %>% filter(system_type=='SIPS')
spis_database <- spis_database %>% mutate(luminary = as.numeric(luminary))

# SPIS  no storage based on system configuration in NPR and USD
spis_database_nostorage <- spis_database  %>% filter (system_configuration =='pv')

spis_database_nostorage %>%
  ggplot (aes(pv_kWp,`system_cost_npr_per_kwp`))+
  geom_point(aes(color=system_configuration),size =3)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'PV Capacity (kWp)', y = 'Cost per kWp NPR',
       color = 'System Configuration',
       title = 'Solar Irrigation Pump Systems - No storage')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
```

```r
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ log10(x), method ='rq',
                colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
                formula = y ~ log10(x), method ='rq',
                colour = '#F8766D', alpha = 0.7, size = 1.5)+
  scale_y_continuous(labels= comma,
                     sec.axis = sec_axis( trans=~.*0.0077,
                                          name="Cost per kWp (USD)", labels =comma)) +
  scale_color_discrete(
    limits = c( "pv"),
    labels = c( "PV without Storage"))

ggsave("sips_scatter_price_perkwp_no_storage.png")
rqfit <- rq(system_cost_npr_per_kwp ~ log10(pv_kWp),
            data = spis_database_nostorage)
summary(rqfit)

# SPIS  with storage based on system configuration in NPR and USD
spis_database_storage <- spis_database  %>%
  filter (system_configuration =='pv_storage')

spis_database_storage %>%   ggplot (aes(pv_kWp,`system_cost_npr_per_kwp`))+
  geom_point(aes(color=system_configuration),size =3)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'PV Capacity (kWp)', y = 'Cost per kWp NPR',
       color = 'System Configuration',
       title = 'Solar Irrigation Pump Systems - With storage')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ log10(x), method ='rq', colour = '#CD9600',
                alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
                formula = y ~ log10(x), method ='rq',
                colour = '#F8766D', alpha = 0.7, size = 1.5)+
  scale_y_continuous(labels= comma,
                     sec.axis = sec_axis( trans=~.*0.0077,
                                          name="Cost per kWp (USD)", labels =comma))+
  scale_color_discrete(
    limits = c( "pv_storage"),
    labels = c( "PV with Storage"))

ggsave("sips_scatter_price_perkwp_storage.png")

rqfit <- rq(system_cost_npr_per_kwp ~ log10(pv_kWp), data = spis_database_storage)
summary(rqfit)

# SPIS  with and without storage based on system configuration in NPR and USD
spis_database%>%   ggplot (aes(pv_kWp,`system_cost_npr_per_kwp`))+
  geom_point(aes(color=system_configuration),size =3)+
  labs(x= 'PV Capacity (kWp)', y = 'Cost per kWp NPR',
```

```
       color = '', title = 'Solar Irrigation Pump Systems')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  scale_x_discrete(labels = c('PV without Storage','PV with Storage'))+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                                        name="Cost per kWp (USD)",
                                                        labels =comma))+
  scale_color_discrete(
    limits = c("pv", "pv_storage"),
    labels = c("PV without  Storage", "PV with Storage")
  )
ggsave("sips_scatter_price_perkwp_all_systems.png")
#


# SHS - Solar Home Systems -----------------------------------------------------

shs_database <- pv_database %>% filter(system_type=='SHS') %>% filter(pv_kWp <= 0.16)

# SHS based on system configuration only AC systems
shs_database  %>% filter(pv_kWp <= 10) %>% filter(system_cost_npr_per_kwp < 1000000) %>%
  ggplot (aes(pv_kWp,`system_cost_npr_per_kwp`))+
  geom_point(aes(color=country),size =3)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'PV Capacity (kWp)', y = 'Cost per kWp NPR',
       color = 'Country', title = 'Solar Home Systems')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                     name="Cost per kWp (USD)", labels =comma))
ggsave("shs_scatter_all_types.png")


# SHS boxplot per country
shs_database  %>%  filter(system_cost_npr_per_kwp < 1000000) %>%
  ggplot (aes('',`system_cost_npr_per_kwp`))+
  geom_boxplot(aes())+
  # stat_summary(fun.y="mean", color="red", shape=15)+
  geom_jitter(aes(size = pv_kWp, color = country), alpha = 0.5)+
  labs(x= '', y = 'System Cost NPR per kWp',
       color = 'Country', size = 'PV (kWp)',  title = 'Solar Home Systems', alpha = NA)+
  theme_bw(base_size = 16)+
  theme(legend.position="right")+
  theme(legend.text=element_text(size=16))+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                 name="Cost per kWp (USD)", labels =comma))
```

```r
ggsave("shs_boxplot.png")

#Calculate median values for SHS
shs_database_means_npr <- shs_database %>% group_by(system_configuration) %>%
  summarise_each(funs(max, min, mean, median),
                 system_cost_npr_per_kwp, system_cost_usd_per_kwp)

# Solar Residential Rooftop ----------------------------------------------------

srt_database <- pv_database %>% filter(system_type =='SRT') %>%
  filter(pv_kWp > 0.15) %>% filter(pv_kWp <= 10) %>%
  filter(system_cost_npr_per_kwp < 500000)

# SRT based on system configuration only AC systems
srt_database  %>% filter(pv_kWp <= 10) %>%
  filter(system_cost_npr_per_kwp < 500000) %>%
  filter(system_configuration != 'pv_dc_couple') %>%
  filter(system_configuration != 'pv_dc_storage') %>%
  ggplot (aes(pv_kWp,`system_cost_npr_per_kwp`))+

  geom_point(aes(color=system_configuration),size =3)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'PV Capacity (kWp)', y = 'Cost per kWp NPR',
       color = 'System Configuration', title = 'Residential Solar Rooftop')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ (x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ (x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma)
#
# ggsave("shs_scatter_only_ac.png")

# Boxplot depending on system configuration (only AC systems)
srt_database  %>% filter(system_configuration != 'pv_dc_couple') %>%
  filter(system_configuration != 'pv_dc_storage') %>%
  ggplot (aes(system_configuration,`system_cost_npr_per_kwp`))+

  geom_boxplot(aes())+
  # stat_summary(fun.y="mean", color="red", shape=15)+
  geom_jitter(aes(size = pv_kWp, color = country), alpha = 0.5)+
  labs(x= '', y = 'Cost per kWp NPR',
       color = 'Country', size = 'PV (kWp)',  title = 'Residential Solar Rooftop',
       alpha = NA)+
  theme_bw(base_size = 16)+
  scale_x_discrete(labels = c('PV without Storage','PV with Storage '))+
  theme(legend.position="right")+
  theme(legend.text=element_text(size=16))+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                     name="Cost per kWp (USD)", labels =comma))
```

```r
ggsave("srt_boxplot_by_system_type.png")

#Calculate median values for SRT with and without storage
srt_database_means_npr <- srt_database %>% group_by(system_configuration) %>%
  summarise_each(funs(max, min, mean, median), system_cost_npr_per_kwp,
                 system_cost_usd_per_kwp)

# Solar Streetlights - SSL --------------------------------------------------

ssl_database <- pv_database %>% filter(system_type=='SSL') %>% filter(luminary >= 20)
ssl_database <- ssl_database %>%  mutate(luminary = as.numeric(luminary))
# Boxplot depending on system configuration

ssl_database  %>%  ggplot (aes(system_configuration,total_cost_npr))+

  geom_boxplot(aes())+
  geom_jitter(aes(size = luminary,), color = '#00BFC4', alpha = 0.5)+
  labs(x= 'Solar Street Light Type', y = 'Cost per Street Light NPR'
       , size = 'LED Lamp Power',  title = 'Solar Street Lights', alpha = NA)+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  scale_x_discrete(labels = c('Type 2','Type 3','Type 4'))+
  # stat_summary(fun.y="mean", color="red", shape=15, size =1)+
  theme(legend.text=element_text(size=16))+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,

                                      name="Cost per kWp (USD)", labels =comma))
ggsave("SSL_cost_per_lamp.png")

ssl_database_means_npr <- ssl_database %>% group_by(system_configuration) %>%
  summarise_each(funs(max, min, mean, median), total_cost_usd, total_cost_npr)

# # Boxplot  depending on Country
# ssl_database  %>%  ggplot (aes(country,`system_cost_npr_per_kwp`))+
#
#   geom_boxplot(aes())+
#   geom_jitter(aes(size = pv_kWp, color = country), alpha = 0.5)+
#   labs(x= 'System Configuration', y = 'Cost per Street Light NPR',
#        color = 'Country', size = 'PV (kWp)',  title = 'Solar Street Lights', alpha = NA)+
#   theme_bw(base_size = 16)+
#   theme(legend.position="right")+
#   theme(legend.text=element_text(size=16))+
#   scale_y_continuous(labels= comma)

# C&I Solar Systems ------------------------------------------------------------

cis_database <- pv_database %>% filter(system_type=='CIS')


cis_database <- cis_database %>% mutate(CIS_scenario = case_when(
  pv_kWp < 90 ~ 'Small C&I (< 100 kWp)',
  pv_kWp >= 90 ~ 'Large C&I ( > 100 kWp)'
))
```

```
# # Boxplot Commercial and Industrial Solar Systems under 300 kWp by country
# cis_database  %>% filter(pv_kWp < 400) %>%
#   ggplot (aes(CIS_scenario ,`system_cost_usd_per_kwp`))+
#   geom_boxplot(aes())+
#   geom_jitter(aes(color = country, na.rm = TRUE), alpha = 0.5)+
#   labs(x= '', y = 'Cost per kWp (USD)',
#        color = 'Country',  title = 'Commercial & Industrial Solar PV')+
#   theme_bw(base_size = 16)+
#   theme(legend.position="right")+
#   theme(legend.text=element_text(size=16))
# ggsave("cis_boxplot_cost_per_kWp_per_size_USD.png")

# Boxplot Commercial and Industrial Solar Systems under 300 kWp by country
cis_database  %>% filter(pv_kWp < 400) %>%
  ggplot (aes(CIS_scenario ,system_cost_npr_per_kwp))+
  geom_boxplot(aes())+
  geom_jitter(aes(color = country, na.rm = TRUE), alpha = 0.5)+
  labs(x= '', y = 'Cost per kWp (NPR)',
       color = 'Country',  title = 'Commercial & Industrial Solar PV')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                     name="Cost per kWp (USD)", labels =comma))

ggsave("cis_boxplot_cost_per_kWp_per_size_USD.png")

# Scatter Commercial and Industrial Solar Systems under 300 kWp
cis_database  %>% filter(pv_kWp < 400) %>%  ggplot (aes(pv_kWp,`system_cost_npr_per_kwp`))+
  geom_point(aes(color=system_configuration),size =3)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'PV Capacity (kWp)', y = 'Cost per kWp NPR',
       color = 'System Configuration', title = 'Commercial and Industrial Systems')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma)

ggsave("cis_scatter_cost_per_kWp.png")

rqfit <- rq(system_cost_npr_per_kwp ~ log10(pv_kWp), data = cis_database)
summary(rqfit)

# calculate means and medians
cis_database_means_npr <- cis_database %>% group_by(CIS_scenario) %>%
  summarise_each(funs(max, min, mean, median), system_cost_npr_per_kwp,
                 system_cost_usd_per_kwp)

# # Boxplot Commercial and Industrial Solar Systems under 300 kWp by country
```

```r
# cis_database  %>% filter(pv_kWp < 400) %>%
#   ggplot (aes(country ,`system_cost_usd_per_kwp`))+
#   geom_boxplot(aes())+
#   geom_jitter(aes(color = country, na.rm = TRUE), alpha = 0.5)+
#   labs(x= '', y = 'Cost per kWp',
#        color = 'Country',  title = 'Commercial & Industrial Solar PV')+
#   theme_bw(base_size = 16)+
#   theme(legend.position="right")+
#   theme(legend.text=element_text(size=16))
# ggsave("CIS_cost_per_kwp_usd_boxplot_country.png")
#
# # Boxplot Commercial and Industrial Solar Systems under 300 kWp
# cis_database  %>% filter(pv_kWp < 400) %>%
#   ggplot (aes('',`system_cost_usd_per_kwp`))+
#   geom_boxplot(aes())+
#   geom_jitter(aes(color = country), alpha = 0.5)+
#   labs(x= '', y = 'Cost per kWp',
#        color = 'Country',  title = 'Commercial & Industrial Solar PV')+
#   theme_bw(base_size = 16)+
#   theme(legend.position="right")+
#   theme(legend.text=element_text(size=16))
# ggsave("CIS_cost_per_kwp_usd_boxplot.png")



# Minigrids ---------------------------------------------------------------------

# Import excel
minigrids_database <- read_xlsx(here::here("data", "system_sizing.xlsx"),
                                col_types = "text")
minigrids_database <- minigrids_database %>% mutate_at(c(6:15),as.numeric)


minigrids_database <- minigrids_database %>% mutate(capex = as.numeric(capex))
minigrids_database <- minigrids_database %>% mutate(pv_kwp = as.numeric(pv_kwp))
minigrids_database <- minigrids_database %>% mutate(cost_per_kwp = capex/pv_kwp)
minigrids_database <- minigrids_database %>% mutate(cost_per_conn = capex/total_conn)
minigrids_database <- minigrids_database %>% mutate(re_scenario = case_when(
  renewable_fraction >= 0.8 ~ 'High RE%',
  renewable_fraction >= 0.6 ~ 'Medium RE%',
  renewable_fraction >= 0.3 ~ 'Low RE%'

))

minigrids_database$re_scenario <- factor(minigrids_database$re_scenario,
                                         levels =c('Low RE%','Medium RE%','High RE%'))


minigrids_database <- minigrids_database %>%
  mutate(distr_per_conn = distribution_capex/total_conn)
minigrids_database <- minigrids_database %>%
  mutate(distr_per_km = distribution_capex/new_line)
minigrids_database <- minigrids_database %>%
```

```r
  mutate(conn_per_kwp = total_conn/pv_kwp)

#Check connections per kWp installed
minigrids_database_100RE <- minigrids_database %>% filter(renewable_fraction >= 0.9)
minigrids_database_100RE  %>% filter(re_scenario =='High RE%') %>%
  filter(pv_kwp <= 600) %>%
  ggplot (aes('',conn_per_kwp))+
  geom_boxplot(aes())+
  geom_jitter(aes(size = pv_kwp, color = country, na.rm = TRUE), alpha = 0.5)+
  labs(x= '', y = 'Connections per kWp installed',
       color = 'Data', size = 'PV (kWp)',
       title = 'Connections per kWp Installed', alpha = NA)+
  theme_bw(base_size = 16)+
  theme(legend.position="right")+
  theme(legend.text=element_text(size=16))
# ggsave("minigrids_connections_per_kwp_installed.png")


# Boxplot for price per kWp based on RE% ber type of data (VS REFERENCE)
minigrids_database <- minigrids_database %>% filter(renewable_fraction > 0.35)
minigrids_database  %>% filter(re_scenario =='High RE%') %>%
  filter(pv_kwp <= 600) %>%
  ggplot (aes( data_gathered,cost_per_kwp, na.rm = TRUE))+

  geom_boxplot(aes())+
  geom_jitter(aes(size = pv_kwp, color = data_gathered, na.rm = TRUE), alpha = 0.5)+
  labs(x= 'Renewable Energy Fraction Scenarios', y = 'Cost per kWp',
       color = 'Data', size = 'PV (kWp)',
       title = 'Solar Minigrids Comparison to AEPC Costs for High RE% Systems', alpha = NA)+
  theme_bw(base_size = 16)+
  theme(legend.position="right")+
  theme(legend.text=element_text(size=16))
# ggsave("minigrids_aepc_high_RE_comparison_boxplot.png")


# Remove AEPC references from minigrids database -------------------------------
minigrids_database <- minigrids_database %>% filter(data_gathered!='AEPC Reference')


# IMAS Sites shown to convey RE% increase drives CAPEX up
mgs_nigeria <- minigrids_database %>% filter(country =='Nigeria')
mgs_nigeria %>% ggplot(aes(renewable_fraction, capex))+
  geom_line(aes(color = site), size = 2)+
  labs(x= 'Renewable Fraction %', y = 'Total Cost USD',
       color = 'Site', title = 'CAPEX vs RE Fraction %', size = "Connections")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_Re_fraction_vs_CAPEX_Nigeria.png")

# # CAPEX vs Connections
# minigrids_database %>% ggplot(aes(total_conn, capex))+
```

```
#   geom_point(aes(color=renewable_fraction, size = pv_kwp))+
#   # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
#   labs(x= 'Total Connections', y = 'Total CAPEX (USD)',
#        color = 'RE Fraction', title = 'SMGs CAPEX vs Connections', size = "PV(kWp)")+
#   theme_bw(base_size = 16)+
#   theme(legend.position="bottom")+
#   theme(legend.text=element_text(size=8))+
#   stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
#                 formula = y ~ log10(x), method ='rq', colour = '#CD9600',
#                 alpha = 0.5, size = 1)+
#   stat_quantile(geom="quantile", quantiles = c(0.5),
#                 formula = y ~ log10(x), method ='rq', colour = '#F8766D',
#                 alpha= 0.5, size = 1.5)+
#   scale_color_gradient(low= "red", high ="green", labels =percent)+
#   theme(legend.key.size = unit(1, 'cm'))+
#   scale_y_continuous(labels= comma)
# ggsave("minigrids_connections_vs_capex_all_RE.png")

# CAPEX vs COnnections high RE (minimum 75%)
# minigrids_database %>% filter(renewable_fraction >= 0.75) %>%  ggplot(aes(total_conn, capex))+
#   geom_point(aes(size = pv_kwp), color = )+
#   # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
#   labs(x= 'Total Connections', y = 'Total CAPEX (USD)',
#        color = 'RE Fraction', title = 'SMGs CAPEX vs Connections (min. 75% RE)', size = "PV(kWp)")+
#   theme_bw(base_size = 16)+
#   theme(legend.position="bottom")+
#   theme(legend.text=element_text(size=8))+
#   stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
#                 formula = y ~ log10(x), method ='rq', colour = '#CD9600',
#                 alpha = 0.5, size = 1)+
#   stat_quantile(geom="quantile", quantiles = c(0.5),
#                 formula = y ~ log10(x), method ='rq', colour = '#F8766D',
#                 alpha= 0.5, size = 1.5)+
#   scale_color_gradient(low= "red", high ="green", labels =percent)+
#   theme(legend.key.size = unit(1, 'cm'))+
#   scale_y_continuous(labels= comma)
# ggsave("minigrids_connections_vs_capex_75RE.png")

# CAPEX vs COnnections high RE (minimum 100%)
minigrids_database %>% filter(renewable_fraction>= 0.9) %>%
  ggplot(aes(total_conn, capex))+
  geom_point(aes(size = pv_kwp), color = '#00BFC4')+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Total Connections', y = 'Total CAPEX (USD)',
       color = 'RE Fraction',
       title = 'SMGs CAPEX vs Connections (+90%  RE)', size = "PV(kWp)")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ (x), method ='rq', colour = '#CD9600',
                alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
```

```
                formula = y ~ (x), method ='rq', colour = '#F8766D',
                alpha= 0.5, size = 1.5)+
  scale_color_gradient(low= "red", high ="green", labels =percent)+
  scale_y_continuous(labels= comma)

ggsave("minigrids_connections_vs_capex_100RE.png")

rqfit <- rq(capex ~ (total_conn), data = minigrids_database)
summary(rqfit)

# rqfit <- rq(system_cost_npr_per_kwp ~ log10(pv_kWp), data = minigrids_database)
# summary(rqfit)

#Plot price per conn. based on size and RE%

minigrids_database  %>%  ggplot (aes(pv_kwp,cost_per_conn))+
  geom_point(aes(color=renewable_fraction, size = pv_kwp))+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'PV (kWp)', y = 'Cost per Connection (USD)',
       color = 'RE Fraction', title = 'Solar Minigrids Cost per Connection vs RE%', size = "PV(kWp)")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ log10(x), method ='rq', colour = '#CD9600',
                alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
                formula = y ~ log10(x), method ='rq', colour = '#F8766D',
                alpha= 0.5, size = 1.5)+
  scale_color_gradient(low= "red", high ="green", labels =percent)+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)

ggsave("minigrids_cost_per_connection_scatter.png")



# Boxplot for price per kWp based on RE%
minigrids_database <- minigrids_database %>%
  filter(renewable_fraction > 0.35) %>%  filter(pv_kwp < 610)

minigrids_database  %>% filter(data_gathered== 'Market Research') %>%
  ggplot (aes( re_scenario,cost_per_kwp))+
  geom_boxplot(aes())+
  geom_jitter(aes(size = pv_kwp),color = '#00BFC4', alpha = 0.5)+
  labs(x= 'Renewable Energy Fraction Scenarios', y = 'Cost per kWp (USD)',
       color = '', size = 'PV (kWp)',  title = 'Solar Minigrids Unit Cost per kWp', alpha = NA)+
  theme_bw(base_size = 16)+
  theme(legend.position="right")+
  theme(legend.text=element_text(size=16))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_cost_per_kwp_boxplot_by_country_RE_category.png")
```

```r
# Boxplot for price per connection based on RE%
minigrids_database <- minigrids_database %>%
  filter(renewable_fraction > 0.35) %>%
  filter(site!='itobe')
minigrids_database  %>%  ggplot (aes( re_scenario,cost_per_conn))+
  geom_boxplot(aes())+
  geom_jitter(aes(size = pv_kwp), color = '#00BFC4', alpha = 0.5)+
  labs(x= 'Renewable Energy Fraction Scenarios', y = 'Cost per Connection (USD)',
       color = 'Country', size = 'PV (kWp)',
       title = 'Solar Minigrids Unit Cost per Connection', alpha = NA)+
  theme_bw(base_size = 16)+
  theme(legend.position="right")+
  theme(legend.text=element_text(size=16))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_cost_per_connection_boxplot_by_country_RE_category.png")

minigrids_database_means_npr <- minigrids_database %>% group_by(re_scenario) %>%
  summarise_each(funs(max, min, mean, median), cost_per_conn, cost_per_kwp)


# Mini grids Distribution Costs ------------------------------------------------

minigrids_database  %>% filter(distr_per_conn > 0) %>%
  ggplot (aes(total_conn,distr_per_conn))+
  geom_point(aes(size = pv_kwp), color ='#00BFC4')+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Total Connections', y = 'Distribution Network Cost per Connection',
       color = 'Re Fraction',
       title = 'Solar Minigrids Distribution Costs per Connection', size = "PV(kWp)")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ (x), method ='rq', colour = '#CD9600',
                alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
                formula = y ~ (x), method ='rq', colour = '#F8766D',
                alpha= 0.5, size = 1.5)+
  scale_color_gradient(low= "red", high ="green", labels =percent)+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_scatter_distr_costs_per_conn_vs_total_conn.png")

# Boxplot in costs per connection distribution
minigrids_database_distr <- minigrids_database %>% filter(distr_per_conn > 0) %>%
  filter(renewable_fraction >= 0.9)
minigrids_database_distr%>% ggplot (aes('',distr_per_conn))+
  geom_boxplot()+
  geom_jitter(aes(size = total_conn, color = total_conn),color ='#00BFC4')+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(size = 'Number of Connections', y = 'Distribution Network Cost per Connection',
       color = 'Re Fraction',x= '', title = 'Solar Minigrids Distribution Costs per Connection')+
  theme_bw(base_size = 16)+
```

```r
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_scatter_distr_costs_per_conn_vs_total_conn.png")

# total distr capex vs lines of distr network
minigrids_database  %>%  ggplot (aes(new_line,distribution_capex))+
  geom_point(aes(size = total_conn),color ='#00BFC4')+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Kilometers of Distribution Network Line', y = 'Distribution Network Cost (USD)',
       color = 'Re Fraction',
       title = 'Solar Minigrids Distribution Costs per km of line', size = "Total Connections")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ (x), method ='rq', colour = '#CD9600',
                alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
                formula = y ~ (x), method ='rq', colour = '#F8766D',
                alpha= 0.5, size = 1.5)+
  scale_color_gradient(low= "red", high ="green", labels =percent)+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_scatter_distr_costs_vs_km_of_line.png")

# distr capex per km of line

minigrids_database  %>% filter(renewable_fraction >=0.8) %>%
  ggplot (aes('',distr_per_km))+
  geom_boxplot()+
  geom_jitter(aes(size = total_conn), alpha = 0.5, color = '#00BFC4')+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= '', y = 'Distribution Network Cost per km of line (USD)',
       color = 'Re Fraction',
       title = 'Solar Minigrids Distribution Costs per km of line', size = "Total Connections")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  scale_color_gradient(low= "red", high ="green", labels =percent)+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_scatter_distr_costs_per_km_of_line.png")

# # cost per km vs total km of distribution network



# Minigrids Distr. Network Connection density vs cost per connection -----------

minigrids_database <- minigrids_database %>% mutate(conn_per_km = total_conn / new_line)
```

```
minigrids_database  %>%  ggplot (aes(distr_per_conn,conn_per_km))+
  geom_point(aes(size = total_conn), color ="#00BFC4" )+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Connections per km of new Distr. Network', y = 'Distribution Network Cost per Connection',
       color = 'Re Fraction', title = 'Solar Minigrids Distribution Costs per km of line', size = "Total
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75),
                formula = y ~ (x), method ='rq', colour ="#CD9600" ,
                alpha = 0.35, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5),
                formula = y ~ (x), method ='rq', colour = '#F8766D',
                alpha= 0.7, size = 1.5)+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)

# distribution network km of new line vs pv kwp -------------------------------

minigrids_database <- minigrids_database %>%
  mutate(adjusted_line_km = new_line *(1 + usable_existing_line))
minigrids_database <- minigrids_database %>%
  mutate(km_per_kwp = adjusted_line_km / pv_kwp)

median(minigrids_database$km_per_kwp, na.rm = TRUE)

minigrids_database %>% ggplot(x='',km_per_kwp)+
geom_boxplot()+
  geom_jitter(aes(size = pv_kWp), alpha = 0.5, color = '#00BFC4')+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= '', y = 'Km of Distr Network per kWp',
      title = 'Solar Minigrids Distribution Costs length per kWP', size = "kWP PV")+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=8))+
  scale_color_gradient(low= "red", high ="green", labels =percent)+
  theme(legend.key.size = unit(1, 'cm'))+
  scale_y_continuous(labels= comma)
ggsave("minigrids_scatter_distr_costs_per_km_of_line.png")


# Components ------------------------------------------------------------------

# Solar Inverters -------------------------------------------------------------

solar_inverter <- pv_database %>%
  filter(system_configuration == 'solar_inverter')

# Solar inverters per countries in NPR
solar_inverter  %>%
  ggplot (aes('',system_cost_npr_per_kwp))+
  geom_boxplot()+
  geom_jitter(aes(color = country), alpha = 0.5)+
```

```
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Solar Inverter Capacity (kW)', y = 'Cost per kW (NPR)',
       color = 'Country', title = 'Solar Inverters', size = 'kW')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile",
                quantiles = c(0.25,0.75), formula = y ~ log10(x), method ='rq',
                colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile",
                quantiles = c(0.5), formula = y ~ log10(x), method ='rq',
                colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                    name="Cost per kWp (USD)", labels =comma))

ggsave("components_solar_inverter_boxplot_npr.png")

# # Solar inverters per countries in USD
# solar_inverter  %>%
#   ggplot (aes(country,system_cost_usd_per_kwp))+
#   geom_boxplot()+
#
#   geom_jitter(aes(color = country), alpha = 0.5)+
#   # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
#   labs(x= 'Solar Inverter Capacity (kW)', y = 'Cost per kW USD',
#        color = 'Country', title = 'Solar Inverters', size = 'kW')+
#   theme_bw(base_size = 16)+
#   theme(legend.position="bottom")+
#   theme(legend.text=element_text(size=16))+
#   stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x), method ='rq', colo
#   stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x), method ='rq', colour = '#
#   scale_y_continuous(labels= comma)

# ggsave("components_solar_inverter_boxplot_usd.png")


# # Solar Inverters depending on size in USD
# solar_inverter   %>% filter(pv_kWp < 400) %>%
#   ggplot (aes(pv_kWp,system_cost_usd_per_kwp))+
#
#   geom_point(aes(color=country))+
#   # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
#   labs(x= 'Solar Inverter Capacity (kW)', y = 'Cost per kW USD',
#        color = 'Country', title = 'Solar Inverters')+
#   theme_bw(base_size = 16)+
#   theme(legend.position="bottom")+
#   theme(legend.text=element_text(size=16))+
#   stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x), method ='rq', colo
#   stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x), method ='rq', colour = '#
#   scale_y_continuous(labels= comma)
#
# solar_inverter_median_npr <- median(solar_inverter$system_cost_npr_per_kwp)
```

```r
# Solar Panels ---------------------------------------------------------------

solar_panels <- pv_database %>% filter(system_configuration == 'pv_panel') %>%
  filter (system_cost_npr_per_kwp>20000) %>%
  filter(system_cost_npr_per_kwp<10000000)

solar_panels  %>%
  ggplot (aes('',system_cost_npr_per_kwp))+
  geom_boxplot()+
  geom_jitter(aes( color = country), alpha = 0.5)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Country', y = 'Cost per kWp NPR',
       color = 'Country', title = 'Solar Panels')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                    name="Cost per kWp (USD)", labels =comma))



# Storage Lithium Ion --------------------------------------------------------

storage_lithium_ion <- pv_database %>%
  filter(system_configuration == 'battery_lithium_ion')

# General average
storage_lithium_ion  %>%
  ggplot (aes(x ='', y = system_cost_npr_per_kwp))+

  geom_boxplot()+
  geom_jitter(aes(color = country), alpha = 0.5)+
  labs(x= 'Battery Storage Lithium Ion', y = 'Cost per kWh NPR',
       color = 'Country', title = 'Storage Lithium Ion')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  # stat_summary(fun.y="mean", color="red", shape=15)+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                    name="Cost per kWp (USD)", labels =comma))

# By country
storage_lithium_ion  %>%
  ggplot (aes(country,system_cost_npr_per_kwp))+
```

```r
  geom_boxplot()+
  geom_jitter(aes(color = country), alpha = 0.5)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Battery Storage Lithium Ion', y = 'Cost per kWh NPR',
       color = 'Country', title = 'Storage Lithium Ion')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma)


# Storage Lead Acid ------------------------------------------------------------

storage_lead_acid <- pv_database %>% filter(system_configuration == 'battery_lead_acid')

# General average
storage_lead_acid  %>%
  ggplot (aes(x ='', y = system_cost_npr_per_kwp))+

  geom_boxplot()+
  geom_jitter(aes(color = country), alpha = 0.5)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Battery Storage Lead Acid', y = 'Cost per kWh NPR',
       color = 'Country', title = 'Storage Lead Acid')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma)

# By country
storage_lead_acid  %>%
  ggplot (aes(country,system_cost_npr_per_kwp))+

  geom_boxplot()+
  geom_jitter(aes(color = country), alpha = 0.5)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Battery Storage Lead Acid', y = 'Cost per kWh NPR',
       color = 'Country', title = 'Storage Lead Acid')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
```

```
  scale_y_continuous(labels= comma)


# PV Mounting ------------------------------------------------------------------


solar_mounting <- pv_database %>%
  filter(system_configuration == 'pv_mounting_rooftop' | system_configuration =='pv_mounting_ground')

solar_mounting  %>%
  ggplot (aes(system_configuration,system_cost_npr_per_kwp))+
  geom_boxplot()+
  geom_jitter(aes(color = country), size = 4 , alpha = 0.5)+
  # geom_line(color='red',data = predicted_conn, aes(x=structures_osm, y=conn_pred))+
  labs(x= 'Solar Mounting (kW)', y = 'Cost per kW NPR',
       color = 'Country', title = 'Solar Mounting Technologies')+
  theme_bw(base_size = 16)+
  theme(legend.position="bottom")+
  theme(legend.text=element_text(size=16))+
  stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
                method ='rq', colour = '#CD9600', alpha = 0.5, size = 1)+
  stat_quantile(geom="quantile", quantiles = c(0.5), formula = y ~ log10(x),
                method ='rq', colour = '#F8766D', alpha = 0.5, size = 1.5)+
  scale_y_continuous(labels= comma, sec.axis = sec_axis( trans=~.*0.0077,
                                    name="Cost per kWp (USD)", labels =comma))+
  scale_x_discrete(labels = c('Ground Mounting','Rooftop Mounting'))

# Get median value for components ---------------------------------------------

components_database <- pv_database %>% filter(system_type =='components')

components_database_means_usdd <- components_database %>%
  group_by(system_configuration) %>%
  summarise_each(funs(max, min, mean, median, sd), system_cost_usd_per_kwp)

components_database_means_usdd <- components_database %>%
  group_by(system_configuration) %>%
  summarise_each(funs(max, min, mean, median, sd), total_cost_usd)

## Error:  <text>:294:17:  unexpected '='
## 293:    stat_quantile(geom="quantile", quantiles = c(0.25,0.75), formula = y ~ log10(x),
## 294:                   =
##                       ^
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
```

```
## locale:
## [1] LC_COLLATE=Spanish_Spain.utf8  LC_CTYPE=Spanish_Spain.utf8
## [3] LC_MONETARY=Spanish_Spain.utf8 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.utf8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] tidylog_1.0.2        scales_1.2.1         quantreg_5.94
##  [4] SparseM_1.81         ggrepel_0.9.1        splitstackshape_1.4.8
##  [7] summarytools_1.0.1   hablar_0.3.0         here_1.0.1
## [10] sjlabelled_1.2.0     forcats_0.5.2        ggplot2_3.3.6
## [13] writexl_1.4.0        readxl_1.4.1         stringr_1.4.1
## [16] tidyr_1.2.1          dplyr_1.0.10         gapminder_0.3.0
## [19] pacman_0.5.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.9         lubridate_1.8.0    lattice_0.20-45    clisymbols_1.2.0
##  [5] rprojroot_2.0.3    digest_0.6.29      utf8_1.2.2         R6_2.5.1
##  [9] cellranger_1.1.0   plyr_1.8.7         backports_1.4.1    MatrixModels_0.5-1
## [13] evaluate_0.16      highr_0.9          pillar_1.8.1       rlang_1.0.5
## [17] data.table_1.14.2  magick_2.7.3       Matrix_1.5-1       checkmate_2.1.0
## [21] labeling_0.4.2     splines_4.2.1      pander_0.6.5       tinytex_0.42
## [25] munsell_0.5.0      compiler_4.2.1     xfun_0.33          pkgconfig_2.0.3
## [29] base64enc_0.1-3    htmltools_0.5.3    tcltk_4.2.1        insight_0.18.3
## [33] tidyselect_1.1.2   tibble_3.1.8       codetools_0.2-18   matrixStats_0.62.0
## [37] fansi_1.0.3        withr_2.5.0        MASS_7.3-57        grid_4.2.1
## [41] gtable_0.3.1       lifecycle_1.0.2    magrittr_2.0.3     cli_3.4.0
## [45] stringi_1.7.8      farver_2.1.1       pryr_0.1.5         reshape2_1.4.4
## [49] rapportools_1.1    generics_0.1.3     vctrs_0.4.1        tools_4.2.1
## [53] glue_1.6.2         purrr_0.3.4        fastmap_1.1.0      survival_3.3-1
## [57] colorspace_2.0-3   knitr_1.40
```

```
Sys.time()
```

```
## [1] "2022-10-07 15:39:59 CEST"
```