# alle Variablen gegeinander geplottet zur Übersicht

# keine Faktorvariablen und ohne Zeit

# NAs (Tage an denen Messungen nicht möglich waren) werden davor entfernt

date\_data\_plot <- date\_data\_noNA[,

c("date", "count\_people",

"ratio", "snow\_diff",

"temperature", "solar\_radiation",

"avalanche\_report",

"cloud\_cover\_daily")] %>%

ggpairs()

## Datum

# Datum und absolute Häufigkeit der Messungen

# NAs (Tage an denen Messungen nicht möglich waren) werden davor entfernt

date\_lvs <- date\_data\_noNA %>%

ggplot(aes(int\_date)) +

geom\_col(aes(y = count\_people, fill = "#F0E442")) +

geom\_col(aes(y = lvs\_true, fill = "darkgreen")) +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr")) +

scale\_fill\_identity(name = " ",

breaks = c("#F0E442", "darkgreen"),

labels = c("Person ohne \n LVS-Gerät", "Person mit \n LVS-Gerät"),

guide = "legend") +

labs(#title = "Checkpointerfassungen in der Wintersaison 18/19 nach Datum",

x = "Datum",

y = "Absolute Häufigkeit")+

theme(text = element\_text(size=20))

ggsave("date\_lvs.png", date\_lvs, dpi = 800, width = 15, height = 10)

# Datum und Ratio

# NAs (Tage an denen Messungen nicht möglich waren) werden davor entfernt

date\_ratio <- ggplot(date\_data) +

geom\_line(aes(int\_date, ratio), color = "darkgreen") +

labs(#title = "Anteil der Personen mit LVS-Gerät in der Wintersaison 18/19",

x = "Datum",

y = "Anteil LVS-Geräte") +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr")) +

scale\_y\_continuous(limits = c(0, 1))+

theme(text = element\_text(size=20))

ggsave("date\_ratio.png", date\_ratio, dpi = 800, width = 15, height = 10)

# Datum und Schneehöhe

date\_snowhight <- ggplot(date\_data) +

geom\_line(aes(int\_date, snowhight)) +

xlab("Datum") +

ylab("Schneehöhe (in cm)") +

scale\_y\_continuous(limits = c(0, 220)) +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr")) +

annotate("rect", xmin = 17903, xmax = 17911, ymin = 0, ymax = 220,

fill = "white", alpha = .8)+

theme(text = element\_text(size=20))

ggsave("date\_snowhight.png", date\_snowhight, dpi = 800, width = 15, height = 10)

# Datum und Schneedifferenz

date\_snowdiff <- ggplot(date\_data) +

geom\_line(aes(int\_date, snow\_diff)) +

xlab("Datum") +

ylab("Schneedifferenz zum Vortag(in cm)") +

scale\_y\_continuous(limits = c(-15, 45)) +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr")) +

annotate("rect", xmin = 17903, xmax = 17911, ymin = -10, ymax = 45,

fill = "white", alpha = .8) +

geom\_hline(yintercept = 0, linetype = 'dotted')+

theme(text = element\_text(size=20))

ggsave("date\_snowdiff.png", date\_snowdiff, dpi = 800, width = 15, height = 10)

# Datum und Temperatur

date\_temperature <- ggplot(date\_data) +

geom\_line(aes(int\_date, temperature))+

xlab("Datum") +

ylab("Temperatur (in °C)") +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr")) +

annotate("rect", xmin = 17903, xmax = 17911, ymin = -8, ymax = 5,

fill = "white", alpha = .8) +

geom\_hline(yintercept = 0, linetype = 'dotted')+

theme(text = element\_text(size=20))

ggsave("date\_temperature.png", date\_temperature, dpi = 800, width = 15, height = 10)

# Datum und Bewölkung

date\_cloud\_cover <- ggplot(date\_data) +

geom\_line(aes(int\_date, cloud\_cover\_daily)) +

xlab("Datum") +

ylab("Bewölkung in %") +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr")) +

annotate("rect", xmin = 17903, xmax = 17911, ymin = 0, ymax = 100,

fill = "white", alpha = .8)+

theme(text = element\_text(size=20))

ggsave("date\_cloud\_cover.png", date\_cloud\_cover, dpi = 800, width = 15, height = 10)

# Datum und Position

date\_position <- ggplot(data[!is.na(data$position),]) +

geom\_bar(aes(int\_date, fill = position), position = "dodge") +

scale\_fill\_manual(values = c("black", "orange"),

name = "Position",

breaks = c("N", "S"),

labels = c("Nord", "Süd")) +

scale\_x\_continuous(breaks = c(17897,17928,17956,17987),

labels = c("01.Jan",

"01.Feb",

"01.Mar",

"01.Apr"))+

labs(#title = "Die Messungen nach Position und Datum",

x = "Datum",

y = "Absolute Häufigkeit") +

theme(text = element\_text(size=20))

ggsave("date\_position.png", date\_position, dpi = 800, width = 15, height = 10)

## Ratio

# NAs (Tage an denen Messungen nicht möglich waren) werden davor entfernt

# Ratio und Schneehöhe

snowhight\_ratio <- date\_data\_noNA %>%

ggplot() +

geom\_point(aes(snowhight, ratio), alpha = 0.5) +

xlab("Schneehöhe (in cm)") +

ylab("Anteil LVS-Geräte")

# Ratio und Temperatur

temperature\_ratio <- date\_data\_noNA %>%

ggplot() +

geom\_point(aes(temperature, ratio), alpha = 0.5) +

xlab("Temperatur (in °C)") +

ylab("Anteil LVS-Geräte")

# Ratio und Bewölkung

cloud\_cover\_ratio <- date\_data\_noNA %>%

ggplot() +

geom\_point(aes(cloud\_cover\_daily, ratio), alpha = 0.5) +

xlab("Bewölkung in %") +

ylab("Anteil LVS-Geräte")

# Ratio und Lawinenwarnstufe

avalanche\_ratio <- date\_data\_noNA %>%

ggplot() +

geom\_jitter(aes(avalanche\_report, ratio), alpha = 0.5) +

labs(x = "Lawinenwarnstufe",

y = "Anteil LVS-Geräte")

## andere Variablen

# Bewölkung

boxplot\_cloud\_cover <- date\_data\_noNA %>%

ggplot() +

geom\_boxplot(aes(y = cloud\_cover\_daily)) +

labs(x = "Bewölkung",

y = "durchschnittliche Bewölkung pro Tag (in %)") +

theme(axis.text.x = element\_blank(), # Entfernt unnötige Zahlen auf der x-Achse

axis.ticks.x = element\_blank())

# Lawinenwarnstufe

boxplot\_avalanche\_report <- date\_data\_noNA %>%

ggplot() +

geom\_boxplot(aes(y = avalanche\_report)) +

labs(x = "Lawinenwarnstufe",

y = "Lawinenwarnstufe jedes Tages") +

theme(axis.text.x = element\_blank(), # Entfernt unnötige Zahlen auf der x-Achse

axis.ticks.x = element\_blank())+

theme(text = element\_text(size=20))

ggsave("boxplot\_avalanche\_report.png", boxplot\_avalanche\_report, dpi = 800, width = 15, height = 10)

# Ferientag

holiday\_plot <- data\_noNA %>%

ggplot() +

geom\_bar(aes(x = holiday, fill = lvs)) +

scale\_fill\_manual(name = " ",

values = c("#F0E442", "darkgreen"),

labels = c("Person ohne \n LVS-Gerät", "Person mit \n LVS-Gerät"),

guide = "legend") +

labs(x = "Ferientag",

y = "Absolute Häufigkeit")+

theme(text = element\_text(size=20))

ggsave("holiday\_plot.png", holiday\_plot, dpi = 800, width = 15, height = 10)

# Wochentag

day\_plot <- data\_noNA %>%

ggplot() +

geom\_bar(aes(x = day, fill = lvs)) +

scale\_fill\_manual(name = " ",

values = c("#F0E442", "darkgreen"),

labels = c("Person ohne \n LVS-Gerät", "Person mit \n LVS-Gerät"),

guide = "legend") +

labs(x = "Wochentag",

y = "Absolute Häufigkeit") +

theme(text = element\_text(size=20))

ggsave("day\_plot.png", day\_plot, dpi = 800, width = 15, height = 10)

## Uhrzeit

# NAs (Tage an denen Messungen nicht möglich waren) werden entfernt

data\_time\_lvs\_plot <- data[!is.na(data$time),] %>%

mutate(time = as.POSIXct(

strftime(

time, format = "%H:%M:%S"),

format = "%H:%M:%S", tz = "GMT"))

date(data\_time\_lvs\_plot$time) <- as.POSIXct("1899-12-31", tz = "GMT")

time\_lvs <- ggplot() +

geom\_freqpoly(data = data\_time\_lvs\_plot,

aes(time, colour = "#F0E442"), binwidth = 15) +

geom\_freqpoly(data = subset(data\_time\_lvs\_plot, lvs == TRUE),

aes(time, colour = "darkgreen"), binwidth = 15) +

scale\_color\_identity(name = " ",

breaks = c("#F0E442", "darkgreen"),

labels = c("Person ohne \n LVS-Gerät",

"Person mit \n LVS-Gerät"),

guide = "legend") +

scale\_x\_datetime(date\_breaks = "2 hour", date\_labels = "%H:%M") +

labs(title = "Die Messungen nach Uhrzeit",

x = "Uhrzeit",

y = "Absolute Häufigkeit") +

geom\_vline(xintercept = as.POSIXct("1899-12-31 04:00:00", tz = "GMT"))+

theme(text = element\_text(size=20))

ggsave("time\_lvs.png", time\_lvs, dpi = 800, width = 15, height = 10)

plot(acf(time\_model$model$residuals), main = " ", ylim=c(0, 1))

scale\_y\_continuous(limits = c(0, 1))

acf((time\_model$model$residuals), ylim=c(0, 1))

ggsave("smooth\_time\_model.png", smooth\_time\_model, dpi = 800, width = 15, height = 10)

# Einzelplots

Tabelle für das Original, Szenario 1 (Generelle Unterschätzung von 22/%), Szenario 2 (Unterschätzung nach Gruppengrö”se), Szenario 3 (Nächtliche Überschätzung) und Szenario 4 (Unterschätzung nach Temperatur)